Benutzerhandbuch



Version 2.0



Inhaltsverzeichnis

KAPITEL 1

Installation

Die Installation des RayQ–Systems erfordert einen minimalen Aufwand. Dennoch sollte sie nur von fachkundigem Personal durchgeführt werden, um Probleme beim Zusammenspiel der einzelnen Komponenten im Voraus ausschließen zu können.

1.1. Vorbereitung

1.1.1. Kommunikationswege sicherstellen

Die Kommunikation zwischen Client und Server basiert ausschließlich auf dem TCP/IP–Protokoll, weshalb eine TCP/IP – Netzwerkverbindung zwingend notwendig ist. Es wird ein frei wählbarer Port für die Verbindung benötigt. Der RayQ Server schlägt standardmäßig Port 1111 vor, Sie können diese Einstellung jedoch jederzeit ändern — während der Installation oder nachträglich in der Software. Stellen Sie vor der Installation der Clients sicher, dass eine freie Verbindung auf dem gewählten Port möglich ist.

1.1.2. Datenbanken im Netzwerk

Um eine Datenbank in das RayQ–System einbinden zu können, muss auf dem Computer, auf dem der RayQ Server betrieben

werden soll, eine entsprechende Datenbank–Verbindung eingerichtet werden. Für eine direkte Verbinung zu einer ORACLE–Datenbank muss der ORACLE–Client ab Version 8.1.5 auf dem Server installiert sein.

Gleiches gilt für den Zugriff über ODBC: Stellen Sie sicher, dass die notwendigen Treiber installiert und die gewünschten Datenquellen konfiguriert sind.

1.1.3. Cashe – Verzeichnis

Der RayQ Server speichert alle Projekte in einem eigenen Dateisystem unter einem frei definierbaren Laufwerkspfad ab. Von der Verwendung eines Netzwerk–Laufwerks ist aus Gründen der Performance und Ausfallsicherheit abzuraten.

Innerhalb des Caches werden alle verwendeten Daten der Projekte gespeichert. Je nach Umfang und Anzahl der zu verwaltenden Projekte ist daher eine ausreichend große Menge an verfügbaren Speicherplatz sicherzustellen. Da ein einzelnes Projekt schnell eine Größe von mehreren hundert Megabyte erreichen kann, ist eine Minimalgröße von zwei Gigabyte empfohlen.

1.2. Installation

Starten Sie die Datei SETUP.EXE aus dem Installationsverzeichnis des RayQ–Pakets. Ist die Autorun–Funktion im Betriebssystem des PCs aktiviert, startet das Setup–Programm nach Einlegen der Installations–CD automatisch (Abbildung 1.1).

1.2.1. Komponenten auswählen

Wählen Sie im ersten Auswahlfenster (Abbildung 1.2) die Programmteile aus, die Sie installieren möchten. Klicken Sie dazu mit der Maus auf die jeweilige Zeile, so dass sie markiert ist. Die Auswahl ist nicht beschränkt, Sie können mehrere Pakete gleichzeitig installieren.

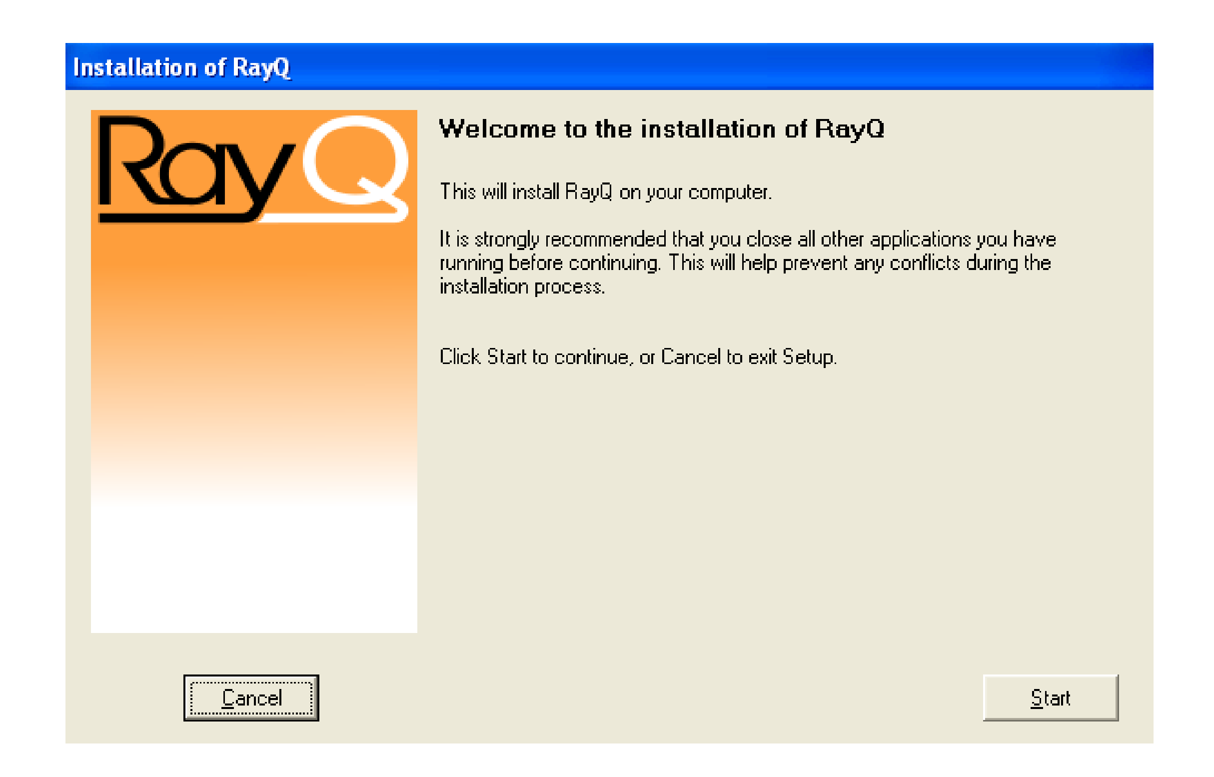


Abbildung 1.1.: Installationsfenster

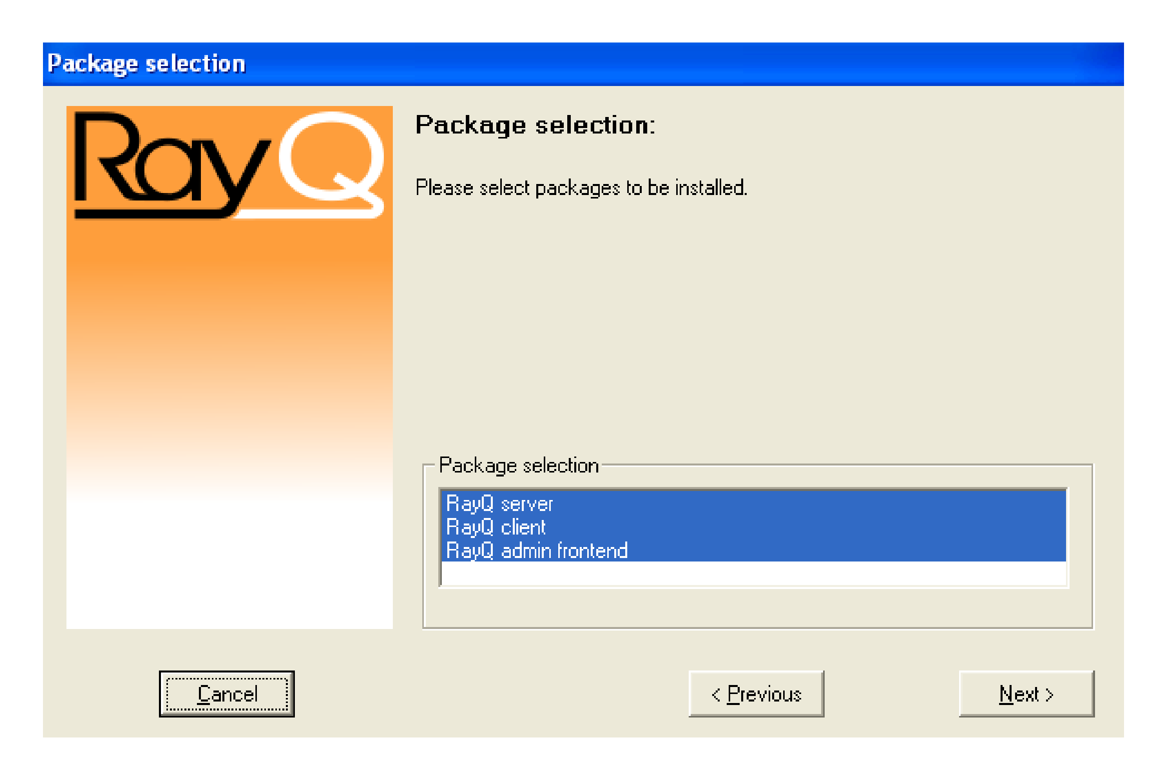


Abbildung 1.2.: Auswahlfenster: Programmteile

1.2.2. Verzeichnisse

Im folgenden Fenster (Abbildung 1.3) haben Sie die Möglichkeit, die Installationspfade abweichend von den Vorgaben zu bestimmen. Im *Installation directory* werden alle ausgewählten Programmkomponenten installiert. Der *Log file path* gibt den Ort an, an dem der RayQ Server seine Protokolldateien speichert. Besonders wichtig ist das Cache directory, da hier die Projektdaten abgelegt werden. Bitte wählen Sie einen Laufwerkspfad, auf dem wie unter 1.1.3 auf Seite 10 erklärt ausreichend Kapazität zur Verfügung steht.



Abbildung 1.3.: Installationspfade

1.2.3. Extra – Parameter der Server – Installation

Administrator–Kennwort

Installieren Sie einen RayQ Server, fordert Sie das Setup – Programm zur Eingabe eines Kennworts für das Administrations – Benutzerkonto auf (Abbildung 1.4). Bitte bestätigen Sie das Kenn-

wort durch zweimalige Eingabe.

Hinweis!
Bei der Installation ist der Name „manager“ als Administrator–Name eingestellt.

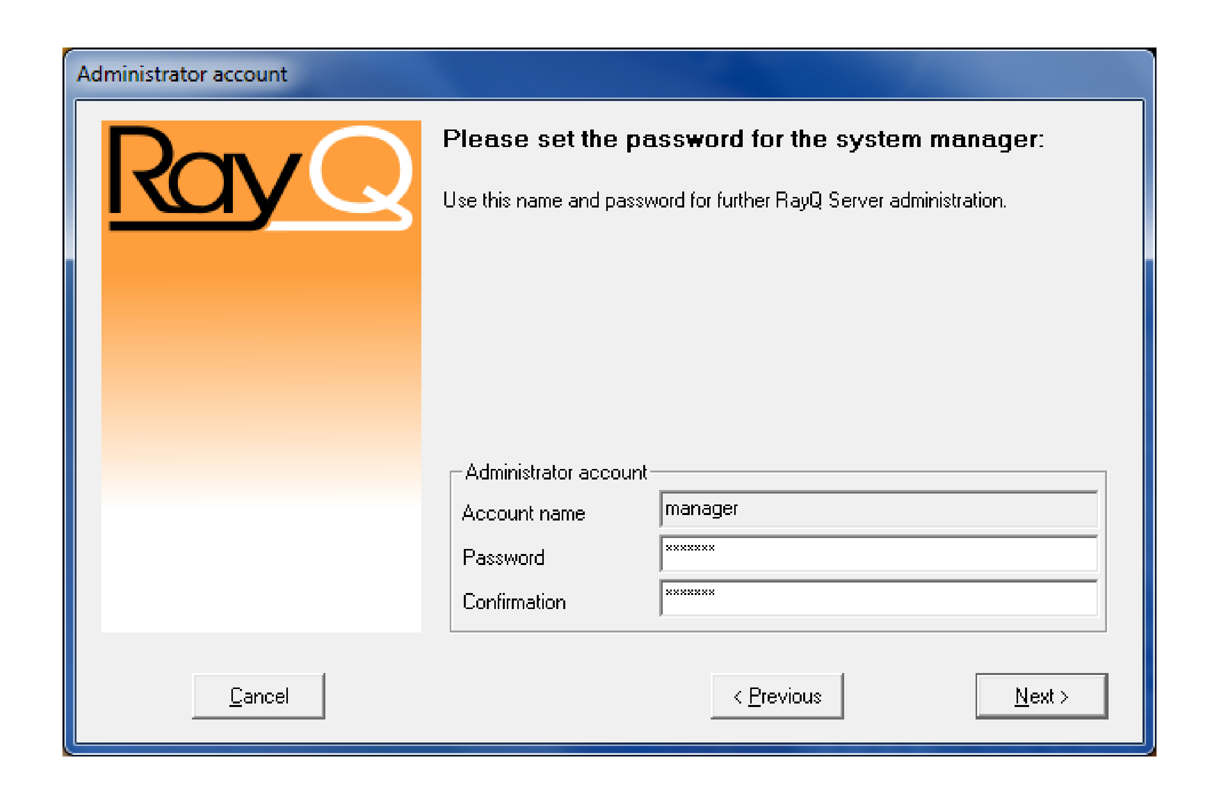


Abbildung 1.4.: Administrations–Benutzerkonto

Datenbank–Treiber

Geben Sie an, welche Datenbanktreiber nach der Installation über den Server zur Verfügung stehen sollen. Zur Auswahl steht ein nativer ORACLE–Datenbanktreiber und ein universeller ODBC–Zugriff. Wählen Sie die gewünschten Treiber durch Anklicken der Zeilen mit der Maus aus.

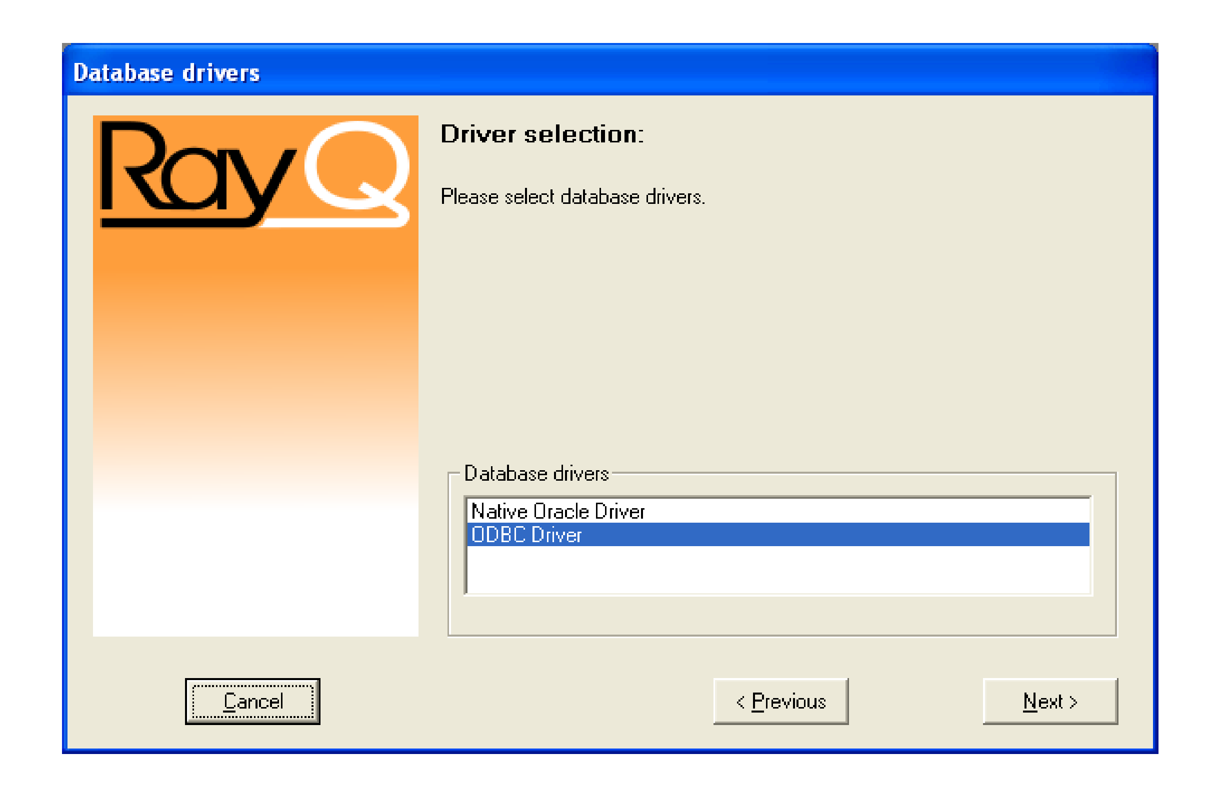


Abbildung 1.5.: Auswahl Datenbank–Treiber

Server–Port

Der RayQ Server benötigt eine Port–Adresse für die Kommunikation zwischen Client und Server. Geben Sie eine im Netzwerk nicht verwendete Port–Adresse ein. Standardmäßig ist der Port 1111 vorgesehen, sie können diese Einstellung jedoch jederzeit ändern.

> **Hinweis!**
> Beachten Sie bitte, dass abweichende und nachträgliche Änderungen bei jedem Client ebenfalls vorgenommen werden müssen.

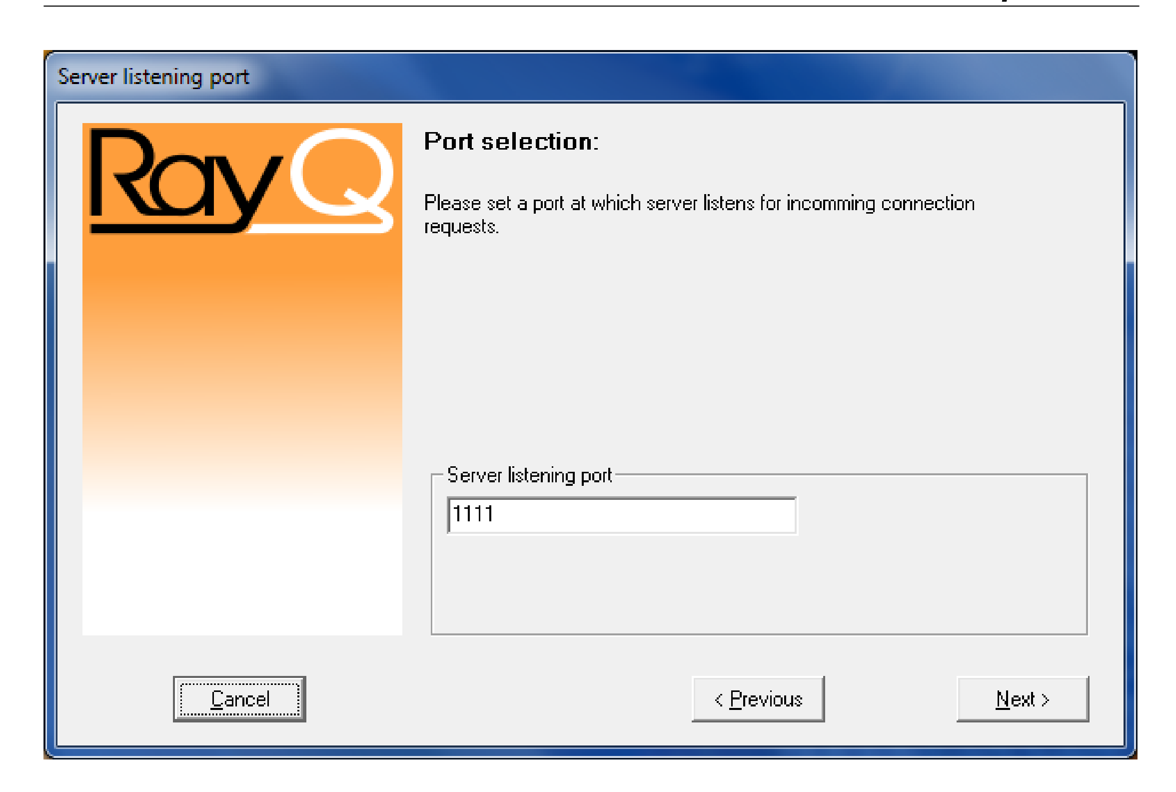


Abbildung 1.6.: Server–Port

1.2.4. Installation starten

Sind alle Daten eingegeben, starten Sie im folgenden Bild (siehe Abbildung 1.7) die Installation. Aus dem Installationsprotokoll entnehmen Sie etwaige aufgetretene Fehler.

> **Hinweis!**
> Führen Sie ein Update durch, müssen alle auf dem betreffenden Rechner laufenden RayQ–Prozesse beendet sein. Andernfalls kann das Setup–Programm einzelne Dateien auf Ihrem System nicht aktualisieren.

1.3. Update

Das Setup–Programm erkennt automatisch, ob bereits eine Version von RayQ auf einem System installiert ist. In diesem Fall wird nur eine Aktualisierung der vorhandenen Daten vorgenommen.

Das Setup–Programm übernimmt die in der Systemkonfiguration gespeicherten Pfade, sowie Kennwort und Adresse des Servers.

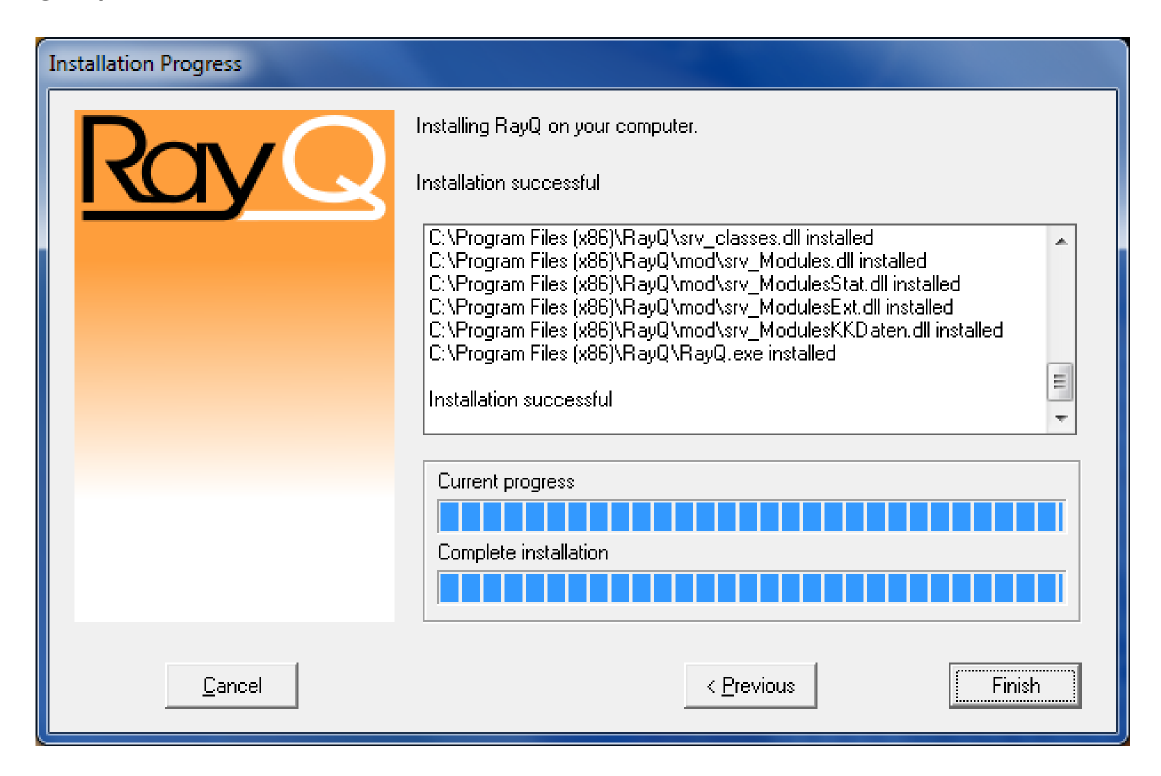


Abbildung 1.7.: Installation starten

Möchten Sie diese Einstellungen nachträglich ändern, verwenden Sie bitte den RayQ Administrator (siehe Seite 217), sowie den Eigenschafts–Dialog des Servers (siehe Seite 235).

1.4. Client–Konfiguration

Die für den Betrieb der Clients (RayQ Designer, RayQ Administrator) notwendigen Einstellungen werden beim ersten Programmstart abgefragt (3.1.1 Server–Adressierung auf Seite 34 und 5.1 Start und Anmeldung auf Seite 217).

1.5. Server als Dienst

Es gibt auch die Möglichkeit den RayQ Administration Server als Dienst zu instalieren. Dazu gehen Sie bitte mit der Kommandozeile in das RayQ–Installationsverzeichnis und geben den Be-

fehl „RQAS–install“ ein. Dadurch wird RayQ als Service installiert, dessen Startparameter wie bei jedem anderen Dienst unter *Systemsteuerung/Verwaltung/Dienste* eingestellt werden können. Mittels des Befehl „RQAS–remove“ kann der Dienst wieder gelöscht werden. Sollten einmal Probleme auftauchen, RQAS zu beenden, so kann mittels des Befehls „RQAS–stop“ eine laufende Instanz von *RQAS.exe* trotz laufender Projekte beendet werden.

1.6. Lizenzdatei

Gemeinsam mit RayQ wurde Ihnen eine Lizenz–Datei mit der Bezeichnung LICENCE.LIC ausgehändigt — entweder als Bestandteil der Installations-CD, auf einem separaten Datenträger oder per E–Mail. Kopieren Sie diese Datei, sofern das nicht bereits geschehen ist, in das Verzeichnis, in das der RayQ–Server installiert wurde. Die Datei enthält alle für den Start des Servers notwendigen Daten über den Umfang und die Art der von Ihnen erworbenen Lizenz.

Wichtig! Ist die Datei nicht vorhanden oder die Lizenz ungültig, so lässt sich der Server nicht starten.

KAPITEL 2

Funktionsprinzip

Das RayQ–System setzt sich aus einzelnen Komponenten zusammen. Im folgenden Kapitel werden wir Ihnen die Funktionsweise des Gesamtsystems, die Rolle der verschiedenen Teile und deren Zusammenspiel erläutern. Eine auf die einzelnen Programmkomponenten eingehende Beschreibung und Hinweise zur Bedienung lesen Sie ab Kapitel 3.

2.1. Architektur der Software

Das RayQ–System teilt sich in zwei große Bereiche auf: In Client und Server. Der Server ist die zentrale Einheit, über die Clients erhalten die Benutzer Zugriff auf das System.

2.1.1. Client

Ein Client ist die Ein- und Ausgabeeinheit des Servers. Ein Client ist ohne Server nicht funktionsfähig, da er keine eigenständige Funktionen enthält, sondern auf die Kommunikation mit dem Server angewiesen ist. Er sendet eine Anforderung an den Server und stellt anschließend dessen Ergebnis dar.

Im RayQ–System stellt der RayQ–Designer den sichtbaren Teil des Clients dar. Mit ihm gestalten Sie die Analyse–Projekte und geben die berechneten Daten aus. Der eigentliche Verarbeitungsprozess findet dabei nur im Server statt. Der Client fungiert quasi als Fernsteuerung und Anzeigeeinheit.

2.1.2. Server

Der Server ist das Gehirn des RayQ–Systems. Er liefert die Rechenleistung, Benutzerverwaltung und führt alle Prozesse durch. Da er keine eigene Ausgabeeinheit besitzt, nutzt er den Client zur Visualisierung der Ergebnisse. Der Server ist ohne Client lauffähig, als permanenter Prozess führt er eigenständig oder auf Anforderung durch den Client die Berechnungen durch. Der Server kann auf alle im Netzwerk verfügbaren Datenbanken zugreifen und den Clients als Datenquelle zur Verfügung stellen.

Die Verwaltung der Projekte findet ausschließlich auf dem Administrations – Server statt. Dadurch spielt es keine Rolle, von welchem Client ein Benutzer auf die Projekte zugreift — durch die zentrale Verwaltung hat er von jedem Ort aus Zugriff.

Mit dem Start eines Clients wird über den Administrations – Server ein neuer Server–Prozess gestartet, der dem jeweiligen Client fest zugeordnet ist. Mit dem Beenden des Clients wird üblicherweise auch der Server–Prozess wieder beendet. Für die Verarbeitung von Analysen im Hintergrund kann der Client auch separat beendet werden, so dass der Server–Prozess weiterhin aktiv bleibt.

2.1.3. Cashe–System

Der Server nutzt ein eigenes Cache–System, das sämtliche Informationen über ein Projekt enthält und zudem alle für die Verarbeitung notwendigen Daten zwischenspeichert. Dadurch wird die Last auf die eigentlichen Datenquellen — in den meisten Fällen eine Datenbank — erheblich reduziert und die Zugriffszeit innerhalb des RayQ–Systems erhöht. Der Client hat über den Server

direkten Zugriff auf den Cache.

Änderungen in einem Projekt werden unverzüglich im Cache gespeichert, so dass auch nach einem System–Ausfall stets der letzte Stand eines Projekts zur Verfügung steht.

2.1.4. Plug–in–Module

Die Einbindung der Rechen- oder Verarbeitungsmodule im System ist über ein Plug–In–Konzept realisiert. Das ermöglicht eine maximale Flexibilität, da für den Austausch oder das Nachrüsten einzelner Module der Server nicht neu installiert oder aufwändig konfiguriert werden muss.

2.1.5. Nachrichtenfluss

Eine Anforderung des Clients wird serverseitig in eine Warteschlange („Queue“) eingereiht. Diese Queue wird von dem zugehörigen Server–Prozess der Reihe nach abgearbeitet. Dabei überprüft er zunächst die Gültigkeit einer Anweisung und führt sie dann aus oder lehnt sie ab. Handelt es sich bei der Anforderung um den Start einer Berechnung, so wird diese Anweisung nach der Überprüfung in eine separate Warteschlange gestellt, die für das Ausführen der Berechnungen zuständig ist. Sie arbeitet weitgehend unabhängig von der Verarbeitung anderer Anforderungen (z.B. Verschieben eines Moduls, Parameter–Änderungen, etc.). Das System verhindert jedoch durch die Gültigkeitsüberprüfung stets, dass die Anweisungen dieser beiden Verarbeitungssysteme miteinander kollidieren. Module, die sich in einer Berechnung oder allgemeinen Änderung befinden, sind für bestimmte Befehle gesperrt.

Alle Ergebnisse werden in einer Nachrichten–Warteschlange an den Client zurück übermittelt, der dann wiederum diese Informationen auswertet und entsprechend darstellt.

Der Client kommuniziert auf diese Weise asynchron mit dem Server. Das bedeutet, dass er nicht unmittelbar auf das Ergebnis ei-

ner Anforderung wartet, sondern eine Anforderung absetzt und schließlich beim Eintreffen des Ergebnisses darauf entsprechend reagiert. Dadurch wird sicher gestellt, dass auch bei länger andauernden Operationen der Client weiterhin ansprechbar bleibt.

Durch die Asynchronität zwischen Client und Server kann es demnach zu Verzögerungen kommen, bis eine angeforderte Änderung auf der Programmoberfläche erscheint. Diese Verzögerung ist vom Umfang der Anforderung, von der momentanen Auslastung des Servers und schließlich der Geschwindigkeit im Netzwerk abhängig. Einfache Anweisungen, wie z.B. das Ändern von Parametern, haben jedoch in der Regel eine Verarbeitungszeit im Millisekunden–Bereich.

2.2. Daten

RayQ ist darauf ausgelegt, mit Datenbanken zu kommunizieren. Deshalb ähnelt die intern verwendete Datenstruktur einer herkömmlichen Datenbank und ist in diesem Bereich universell einsetzbar.

2.2.1. Tabellen und Spalten

Die Daten verwaltet das RayQ–System in Tabellen. Diese setzen sich aus Spalten zusammen, die wiederum aus Zeilen bestehen. Bei Berechnungen werden die Original–Daten niemals verändert, sondern die Ergebnisse in Form von neuen Spalten hinzugefügt oder als neue Tabelle zur Verfügung gestellt.

Eine Tabelle ist an jedem Punkt des RayQ–Systems exakt gleich. Änderungen wirken sich stets projektweit aus: Ändert sich der Inhalt einer Tabelle, die als Quelle in eine Berechnung einbezogen ist, wird ein zuvor ermitteltes Ergebnis mit dieser Änderung ungültig und muss neu berechnet werden.

Markierungen

Zeilen einer Tabelle lassen sich mit Markierungen versehen, was in Views (siehe 4.8 Views auf Seite 194) angezeigt oder als Gruppierungskriterium in einer Berechnung (siehe 4.3.10 Group Mark auf Seite 129) Verwendung findet. Die Markierungen stehen in allen Modulen zur Verfügung und werden auch auf andere Tabellen übertragen, die durch eine Berechnung aus der ursprünglichen Tabelle entstanden sind.

Mithilfe dieser Markierungen lassen sich an verschiedenen Stellen einer Analyse Datensätze identifizieren und auffinden.

2.2.2. Datentypen

Beim Einlesen der Daten aus einer externen Datenquelle werden die Datentypen automatisch an die im RayQ–System verwendeten angepasst.

Das Datenformat einer Spalte ist stets einheitlich, das heißt alle Zeilen einer Spalte besitzen denselben Typ. Des weiteren haben alle Spalten einer Tabelle dieselbe Zeilenzahl.

Mehrdimensionale Spalten

Üblicherweise hat eine Spalte eine einzige Dimension: Zeilen. RayQ ist jedoch in der Lage, Spalten mit mehreren Dimensionen zu verwalten — praktisch eine Tabelle innerhalb einer Tabelle.

Notwendig werden solche Spaltenformen zum Beispiel im Kohonen – Modul (siehe Seite 171), das eine zweidimensionale Matrix als Ergebnis liefert. Diese Matrix kann z.B. als Höhengrafik (siehe 4.8.3 Surface auf Seite 211) oder Tabelle (siehe 4.8.2 Matrix View auf Seite 199) dargestellt werden.

Auch eine Korrelation (siehe Seite 108) liefert als Ergebnis eine zweidimensionale Matrix, die Zuordnungen darstellt. Das Pivot–Modul (siehe Seite 138) verwendet eine besondere Form von

mehrdimensionalen Spalten, die für die besondere Darstellung dieser Tabelle verwendet wird.

Die grundsätzliche Möglichkeit des RayQ–Systems, mehrdimensionale Spalten verarbeiten zu können, ermöglicht eine Vielzahl von Anwendungsfällen, die durch eine klassische Tabellenstruktur nicht oder nur teilweise abgebildet werden können.

2.2.3. Genauigkeit

Die Genauigkeit von numerischen Formaten hängt vom Betriebssystem und der verwendeten Hardwareplattform ab und ist daher begrenzt. Unter Windows/Intel arbeitet RayQ bei Fließkomma–Operationen mit einer maximalen Genauigkeit von 64 Bit. Durch die binäre Codierung der Fließkommazahlen entsteht eine Begrenzung der Genauigkeit auf eine maximale Anzahl von Dezimalstellen und 10er–Potenzen.

Wichtig!
Bitte beachten Sie, dass durch z.B. Transformationen und entsprechende Berechnungen die Genauigkeit der Werte durch die Rundung der letzten Nachkommastellen stetig abnimmt. Dieser Effekt potenziert sich durch mehrere aufeinander folgende Berechnungen dieser Art und kann im ungünstigsten Fall die Ergebnisse unbrauchbar machen. Berücksichtigen Sie diesen Umstand bei Ihren Analysen!

2.3. Module

Ein Modul stellt im RayQ–System den kleinsten Baustein eines Projekts dar. Ein Modul besitzt in der Regel einen Ein- und einen Ausgang, über die Tabellen von anderen Modulen zur Verarbeitung empfangen und ausgegeben werden. Der Datenfluss wird

durch so genannte Links vom Ausgang eines zum Eingang eines zweiten Moduls hergestellt und besteht nur in dieser Richtung. Markierungen innerhalb einer Tabelle stehen auch „rückwärts“ in vorhergehenden Modulen zur Verfügung, da sie keine Veränderung der Tabellendaten darstellen. Ein Modul kann mehrere Links empfangen und zu mehreren anderen Modulen verzweigen.

Ein Modul fügt durch seine Berechnung einer eingehenden Tabelle neue Spalten hinzu oder stellt vollständig neue Tabellen zur Verfügung. Ein Modul kann nur die von ihm generierten Tabellen verändern. Die Inhalt eingehender Tabellen anderer Module bleiben unberührt.

2.3.1. Wirkungsweise von Modulen auf Tabellen

Je nach Art des Moduls

- erzeugt es Tabellen vollständig neu (z.B. „Datasource“).
- stellt eine neue Tabelle aus den Spalten der eingehenden Tabelle und neu erzeugte Spalten zusammen. Diese Tabelle kann beliebig umbenannt werden ohne die eingehende Tabelle zu verändern. Wird eine Spalte der eingehenden Tabelle umbenannt, wird sie gleichzeitig in der eingehenden und der erzeugten Tabelle umbenannt, da es sich logisch um identische Spalten handelt (z.B. Kohonen).
- ersetzt es eine oder mehrere eingehende Tabellen vollständig durch eine oder mehrere neu erzeugte (z.B. Join–Tool).
- leitet es eine eingehende Tabelle ohne Änderungen durch (z.B. Views).
- erzeugt es eine vollständig neue Tabelle, ohne Spalten der eingehenden Tabellen zu übernehmen (z.B. Kohonen).

Alle diese Arten der Änderung können auch miteinander kombiniert werden.

2.3.2. Analysekette

Die Parametrisierung der Berechnung erfolgt stets direkt über die betreffende Spalte einer Tabelle und kann jederzeit verändert werden. Diese permanent gegebene Möglichkeit der Beeinflussung des Ergebnisses durch Anpassen der Parameter ermöglicht Ihnen einen vollständig iterativen Analyseprozess.

> **Hinweis!**
> Änderungen haben in den meisten Fällen Auswirkungen auf nachfolgende Module. Ändern sich die Quelldaten eines Moduls, wird eine bereits vorhandene Ergebnismenge ungültig und muss neu berechnet werden. Beim Starten einer Analyse werden nur die Module mit ungültigen Daten ausgeführt, um doppelte Berechnungen zu vermeiden.

Soll die Verarbeitung eines Moduls gestartet werden, ohne dass gültige Daten vom Vorgängermodul vorliegen, so wird dessen Berechnung zuerst ausgeführt. Diese Kette setzt sich rückwärs beliebig fort, so dass bei einer fertig aufgebauten Analyse immer nur das letzte Modul gestartet werden muss, um ein aktuelles Ergebnis zu erhalten.

Während einer Berechnung sind die betroffenen Module für Änderungen gesperrt.

> **Wichtig!**
> Eine Änderung kann ebenfalls zur Folge haben, dass Berechnungen nachgeschalteter Module nicht mehr möglich sind, wenn sich z.B. Datentypen ändern oder Spalten wegfallen.

Gültigkeit der Daten

Von einer Datenquelle kommende oder von einem Berechnungsmodul erzeugte Daten einer Tabelle sind nach dem Ausführen des jeweiligen Moduls *gültig*. Werden Änderungen an einem Modul durchgeführt, die die ausgegebenen Daten betreffen, so wird diese Tabelle *ungültig*. Beziehen sich weitere Module auf diese Tabelle, so werden auch deren Daten ungültig.

Eine Änderung an der Quelle, also dem Anfang einer Analysekette, markiert alle davon abhängenden Tabellen als ungültig. Eine Berechnung des letzten Moduls einer Analysekette startet die Berechnungen aller Vorgänger–Module mit ungültigen Daten, sodass alle Tabellen dieser Kette anschließend gültig sind.

2.3.3. Modularten

Die in RayQ zur Verfügung stehenden Module lassen sich nach dem EVA–Prinzip — Eingabe, Verarbeitung, Ausgabe — in diese drei Bereiche aufteilen.

> **Hinweis!**
> Eine detaillierte Beschreibung aller im System vorhandenen Module finden Sie ab Seite 63.

Datenquellen

RayQ bietet direkten Zugriff auf ORACLE–Datenbanken sowie auf alle über ODBC erreichbaren Systeme. Ihnen stehen alle Möglichkeiten der SQL–Abfragen zur Verfügung. Zusätzlich lassen sich alle Datenquellen, die dem jeweiligen System über die ADO–Schnittstelle (u.a. ODBC) zur Verfügung gestellt werden, über den RayQ Designer lokal importieren.

Zu Test- und Kontrollzwecken stellt RayQ des weiteren eigene Datenquellen zur Verfügung, die z.B. Zahlengeraden oder Zufalls-

reihen generieren.

Zu der Kategorie der Datenquellen zählen ebenfalls die Module, die zwei oder mehrere Tabellen zu einer neuen zusammenfügen und als neue Datenmenge zur Verfügung stellen.

Die in das RayQ–eigene Datenformat „RQD“ exportierten Tabellen werden ebenfalls als eigenständige Datenquelle aufgeführt. Sie enthalten die zuvor exportierten Daten und sich nachträglich nicht veränderbar (siehe 4.6.2 RayQ Data Export auf Seite 189).

Verarbeitung

Module, die vorhandene Tabellen zur Berechnung neuer Datenreihen verwenden, fallen in die zweite Kategorie. Neben einem Modul, das die freie Eingabe von mathematischen Formeln erlaubt, finden Sie hier noch Weitere, die die gebräuchlichsten Berechnungsformen zur Verfügung stellen.

Das Kohonen–Netzwerk stellt in dieser Gruppe eine eigene Analyseform dar, da die Quelldaten dazu verwendet werden, mithilfe eines Neuronalen Netzes Muster zu erkennen und anhand der dabei entstehenden Karte die Daten zu klassifizieren (siehe 4.4.1 Kohonen auf Seite 171).

Ausgabe

Zur Ausgabe der Tabellen bestehen drei Möglichkeiten: tabellarisch, grafisch oder als Datei–Export. In der grafischen Darstellung lassen sich zudem noch direkte Vergleiche und Zuordnungen mit verschiedenen Graphen herstellen.

Eine besondere Form stellt der Export in das RQD–Format dar, bei dem eine RayQ–Tabelle direkt gespeichert und fortan als eigenständige Datenquelle verwendet werden kann (siehe 4.6.2 RayQ Data Export auf Seite 189).

2.4. Projektverwaltung

RayQ–Projekte werden im servereigenen Cache–System zentral gespeichert, sodass sie jedem Client zur Verfügung stehen. Dieses Prinzip ähnelt dem einer Datenbank: Jede Änderungen an einem Projekt wird nach einer internen Gültigkeitsüberprüfung direkt auf dem Server gespeichert. Dadurch erhält das System eine hohe Ausfallsicherheit, da stets die aktuelle Version des Projekts gesichert ist.

Deshalb gibt es auch keine eigene Funktion zum Speichern eines Projekts in der Benutzeroberfläche — die Daten sind ja bereits gesichert. Als Optionen verbleiben noch, das Projekt mit seinen während der Bearbeitung gesicherten Änderungen auf dem Server zu belassen oder schließlich komplett zu löschen.

2.4.1. Projekt–Verarbeitung im Hintergrund

Mit dem Beenden des Designer–Clients wird im Normalfall auch der jeweils zugeordnete Server–Prozess beendet und das aktuelle Projekt gespeichert oder gelöscht. Für Projekte, die einen längeren, zeit- oder ereignisgesteuerten Prozess bearbeiten, lässt sich der Client separat beenden (siehe 3.7.1 Optionen beim Schließen auf Seite 57).

Der Server–Prozess bleibt dabei weiterhin aktiv und führt seine Berechnungen durch. Wird ein Trigger–Modul (siehe 4.5 Trigger auf Seite 180) verwendet, kann er z.B. auf Datenbank–Ereignisse reagieren und eine Analyse starten.

Wird das Projekt von einem Designer–Client wieder geladen, wird direkt der laufende Server–Prozess wieder aufgegriffen. Das Projekt kann damit z.B. angehalten und weiter bearbeitet werden.

2.4.2. Archivierung

Vorhandene Projekte lassen sich zusätzlich zur normalen Speicherung im Cache–System in ein Archiv speichern. Dabei wer-

den nur Struktur und Parameter der Module gespeichert, nicht aber die Daten. Nutzen Sie diese Funktion, um z.B. Zwischenstände der Arbeit an einem Projekt zu speichern. Beim Öffnen eines solchen Archivs wird ein neues, eigenständiges Projekt angelegt, das dieses Archiv als Vorlage verwendet.

Hinweis!
Möchten Sie nur die Daten einer Berechnung unabhängig vom Projekt abspeichern, verwenden Sie den RayQ Data Export (siehe Seite 189).

2.5. Benutzerverwaltung

RayQ verfügt über eine einfache aber effektive Benutzerverwaltung, die verhindert, dass nicht autorisierte Benutzer das System verwenden. Dabei wird zwischen Benutzern unterschieden, die den Designer–Client verwenden und Analysen durchführen, und Benutzern, die zusätzlich Administrationsrechte besitzen.

Benutzer ohne Administrationsrechte haben vollen Zugriff auf alle Funktionen des Designer–Clients.

Administratoren haben zusätzlich das Recht im Administrations–Frontend weitere Benutzer anzulegen, Datenbanken am System anzumelden, das Ereignisprotokoll und eine Liste aller laufenden und gespeicherten Projekte einzusehen.

2.5.1. Datenbank–Zugriff

Zum Zugriff auf eine Datenbanke muss das jeweilige Datenbank–Passwort verwendet werden. Eine Ausnahme bilden Datenbanken, die automatisch den am Betriebssystem angemeldeten Benutzer authentifizieren.

Wichtig!
Bei vielen Datenbank–Systemen besteht die Möglichkeit, eine Autorisierung über den am Betriebssystem angemeldeten Benutzer durchzuführen. Da RayQ über seinen zentralen Administrations–Server auf die Datenbank zugreift, würde in diesem Fall immer der angemeldete Benutzer des Systems verwendet, auf dem dieser Server betrieben wird — und nicht der Benutzer des RayQ–Clients. Aus diesem Grund ist es ratsam, die Datenbank–Zugriffsverwaltung nur über eine direkte Eingabe von Benutzername und Passwort zu verwenden.

KAPITEL 3

RayQ Designer

Der Designer ist die Schaltzentrale Ihres RayQ–Systems. Mit ihm erzeugen und verwalten Sie alle Projekte, starten Analysen, geben grafische Auswertungen aus oder generieren Tabellen.

Die gesamte Programmoberfläche ist für die Bedienung mit der Maus konzipiert. Per Drag & Drop platzieren Sie neue Module auf ihrem Arbeitsbereich, verbinden sie miteinander und starten die Analyse. Die einzelnen Programmteile sind darauf hin optimiert, Ihnen die Arbeit mit dem System so einfach und intuitiv wie möglich zu gestalten.

Der Designer teilt sich in ein Hauptfenster, View–Fenster und verschiedene Dialoge auf. Für den Betrieb des Designers ist eine Anbindung zu einem RayQ–Server erforderlich. Ohne Server kann der Designer nicht gestartet werden.

3.1. Programmstart

Starten Sie den RayQ–Designer über das Programmsymbol auf Ihrer Windows–Oberfläche.

3.1.1. Server–Adressierung

Ist auf dem System noch keine Verbindung zum RayQ–Server eingerichtet, erscheint nach dem ersten Start des Designers ein Konfigurations–Dialog (Abbildung 3.1), der Sie zur Angabe der Server–Adresse auffordert. Geben Sie dort die IP–Adresse und deren Port ein, unter der der RayQ–Server im Netzwerk erreichbar ist.



Abbildung 3.1.: Server–Einstellungen

Wichtig!
Nach der Eingabe werden diese Daten auf dem Rechner lokal gespeichert, sodass sie der RayQ–Designer beim nächsten Start automatisch verwenden kann. Möchten Sie nachträglich die Server–Adresse ändern, rufen Sie im Menü *File/Server Settings* den Konfigurations–Dialog erneut auf.

3.1.2. Login

Nach dem Start des RayQ–Designers erscheint zunächst ein Anmelde – Dialog (Abbildung 3.2), über den Sie sich im System autorisieren müssen.

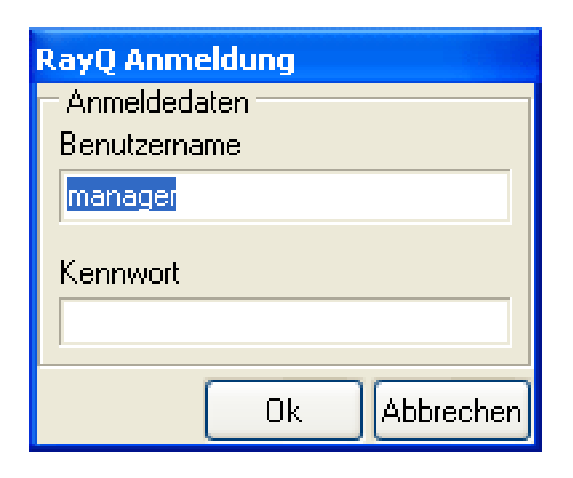


Abbildung 3.2.: RayQ–Anmeldung

Geben Sie hier den im RayQ–Server eingerichteten (siehe 5.2.3 Users auf Seite 221) Benutzernamen und das dazu gehörige Kennwort ein. Der zuletzt verwendete Benutzername wird auf Ihrem Computer gespeichert, so dass er beim nächsten Start automatisch angezeigt wird. Das Kennwort wird aus Sicherheitsgründen nicht gespeichert und bei der Eingabe nur verdeckt angezeigt.

> **Hinweis!**
> Die Verwaltung der Benutzerkonten erfolgt zentral auf dem RayQ Server.

3.1.3. Projektauswahl

Nach dem Start des RayQ Designers öffnet sich der Dialog aus Abbildung 3.3. Klicken Sie auf *OK*, um ein neues Projekt anzulegen oder wählen ein Gespeichertes aus der angezeigten Liste aus. Die Spalte Name gibt den vom Benutzer vergebenen Projektnamen an, *ID* die vom System bei der Erstellung des Projekts vergebene fortlaufende Projektnummer und *Type* den Projekttyp.

3.2. Projektverwaltung

Ihre Analysen werden in einzelnen Projekten vom RayQ–Server unter einem von Ihnen frei wählbaren Namen verwaltet. Mit dem Designer legen Sie neue Projekte an, verändern vorhandene oder starten Analysen.

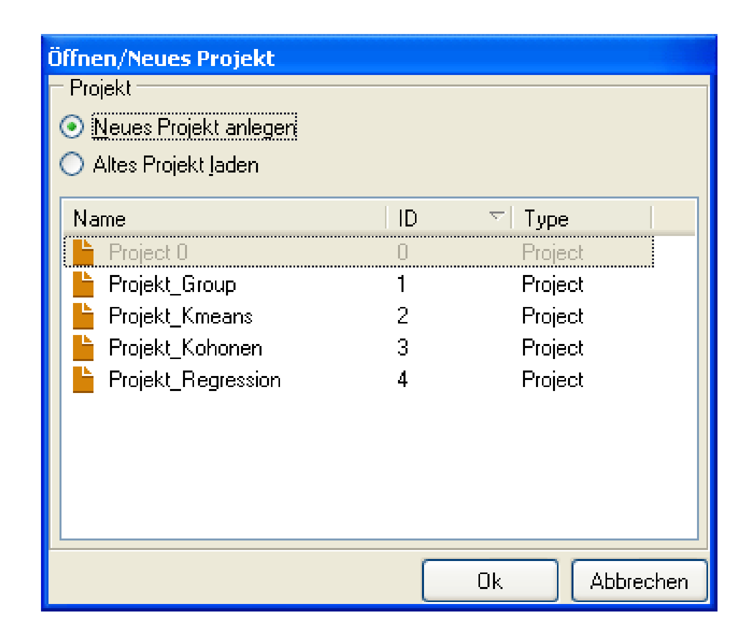


Abbildung 3.3.: Projektverwaltung

Neues Projekt erstellen. Öffnen Sie im Hauptmenü *Datei/Öffnen/Neues Projekt* den Dialog von Abbildung 3.3 und wählen die Option *Neues Projekt anlegen*, um ein neues, leeres Projekt zu öffnen. Der Server vergibt einen generischen Namen, den Sie jederzeit ändern können.

Bestehendes Projekt laden. Die Liste zeigt alle für Sie auf dem RayQ–Server zur Verfügung stehenden Projekte. Wählen sie mit der Maus oder Tastatur das gewünschte aus und bestätigen mit OK.

Namen vergeben. Einem geladenen Projekt können Sie über *Projekt umbenennen* zu einem beliebigen Zeitpunkt einen neuen Namen geben. Dabei wird keine Kopie des Projekts angelegt,

sondern in der Projektverwaltung des Servers der aktuelle Eintrag umbenannt.

Projekt speichern. Ein geöffnetes Projekt wird automatisch im Hintergrund nach jeder vorgenommenen Änderung auf dem Server gespeichert, weshalb eine gesonderte Speicher–Funktion nicht notwendig ist.

Protokoll ausgeben. Für Dokumentationszwecke lässt sich die Konfiguration des aktuellen Projekts in ein Protokoll speichern. Wählen Sie im Hauptmenü *Datei/Protokoll* aus. Die generierte Liste wird im RayQ–Programmverzeichnis unter dem Namen des Projekts als Textdatei abgespeichert. Sie enthält eine Aufzählung aller verwendeten Module, deren Tabellen, Parameter und Verbindungen.

Hinweis!
Dokumentiert werden ausschließlich Informationen über die Struktur des Projekts. Das Protokoll enthält keine Inhalte von Tabellen oder Zugriffskennwörter.

3.3. Hauptfenster

3.3.1. Hauptmenü

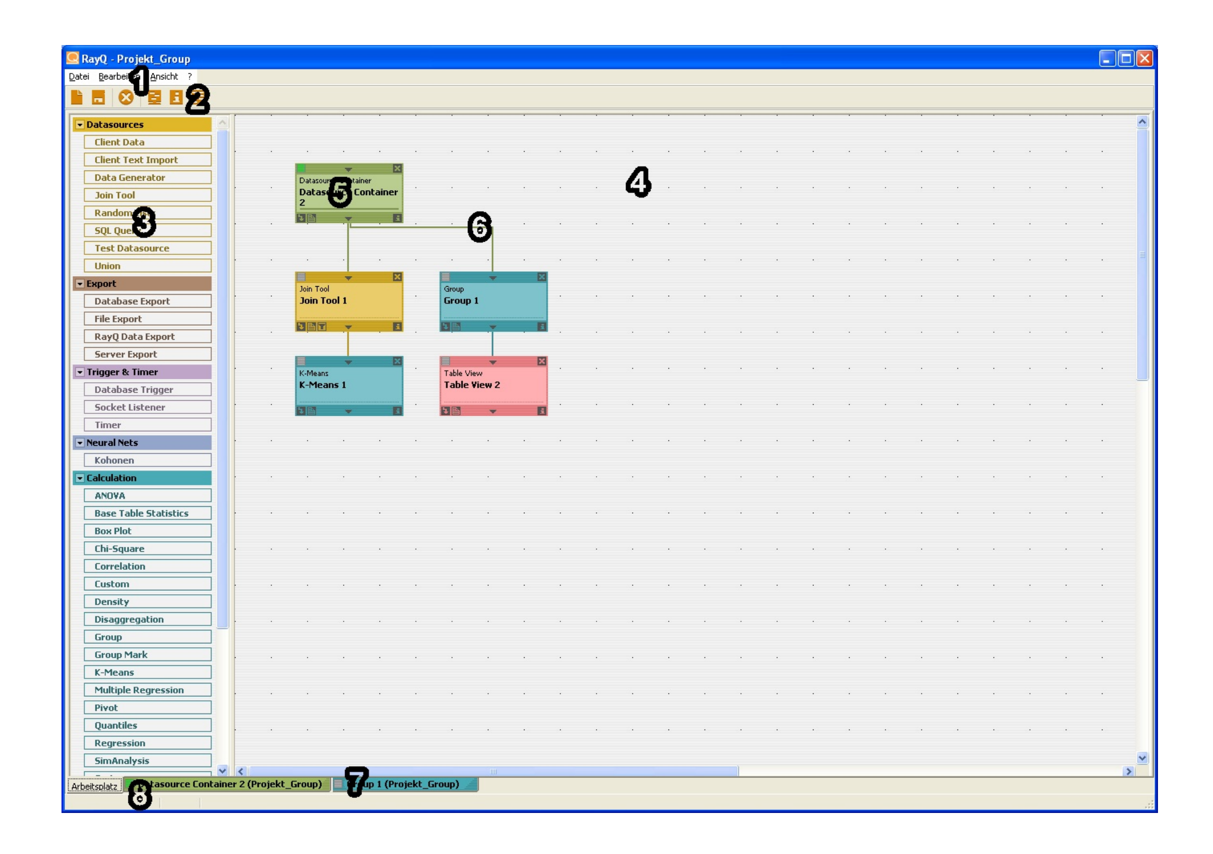


Abbildung 3.4.: Hauptfenster

1. Hauptmenü (siehe Seite 38)
2. Schnellstart–Leiste mit den wichtigsten Menü–Funktionen
3. Modul–Leiste (siehe Seite 49)
4. Workspace (siehe Seite 42)
5. Module (siehe Seite 42)
6. Link (siehe Seite 45)
7. Register (siehe Seite 46)
8. Status–Anzeige

Jedes geladene Projekt wird in einem eigenen Hauptfenster dargestellt und kann unabhängig von anderen bearbeitet und ausgeführt werden. Ist das letzte Hauptfenster geschlossen, wird der RayQ–Designer beendet.

Im Folgenden möchten wir Sie mit den Menüs und den einzelnen Programm–Modulen vertraut machen (Abbildung 3.4).

Menüeinträge „Datei“	Funktion
Öffnen/Neues Projekt	Wählen Sie *Neues Projekt anlegen* zum Anlegen eines neuen, leeren Projekts oder selektieren Sie ein auf dem Server gespeichertes aus der Liste und bestätigen mit OK oder einem Doppelklick auf die jeweilige Zeile. *Siehe 3.2 Projektverwaltung auf Seite 36*
Projekt umbenennen	Neu angelegte Projekte erhalten einen automatisch generierten Namen, unter dem sie zunächst verwaltet und abgelegt werden. Um dem aktuellen Projekt einen neuen Namen zuzuweisen, verwenden Sie die *Umbenennen*–Funktion und geben in dem erscheinenden Eingabefenster einen beliebigen neuen Namen ein. *Siehe 3.2 Projektverwaltung auf Seite 36*
Projektstruktur archivieren	Legt vom aktuellen Projekt eine Kopie im Archiv des Servers ab, die jedoch keine Daten enthält. Verwenden Sie diese Funktion, um z.B. Sicherheitskopien von Zwischenständen Ihrer Arbeit zu sichern. Als Name wird der aktuelle Projektname vorgeschlagen, kann aber von Ihnen vor dem Speichern abgeändert werden. *Siehe 3.2 Projektverwaltung auf Seite 36*

Protokoll	Gibt eine Liste aller im aktuellen Projekt vorhandenen Module und deren Parameter als Textdatei aus. *Siehe Protokoll ausgeben auf Seite 37*
Einstellungen	Im Register *Fenstereinstellungen* lassen sich das Hauptfenster und die Modulansichten anpassen. Die Sprachwahl ist zwischen Deutsch und Englisch möglich, erfordert jedoch einen Neustart. Unter *Module* lassen sich Voreinstellungen laden bzw. löschen. Die Arbeitsplatzfläche lässt sich im Register *Projekt* anpassen und als Standard für neue Projekte zuweisen. *Siehe 3.8 Einstellungen speichern ab Seite 58*
Server Einstellungen	Ermöglicht die Änderung der Verbindungseinstellungen zum RayQ Server. *Siehe 3.1.1 Serveradressierung auf Seite 34*
Change Password	Ermöglicht die Änderung des Passwortes.
Schließen	Schließt das aktuelle Projekt. *Siehe 3.7 Projekt schließen auf Seite 57*

Tabelle 3.1.: Menü „Datei“

Menüeinträge „Bearbeiten“	**Funktion**
Container herstellen	Speichert die im Workspace markierten Module und wandelt sie in einen Container um. *Siehe 4.7 Templates ab Seite 191*

Container Speichern	Legt einen Container auf dem Server ab, um ihn als Template–Modul zur Verfügung zu stellen. *Siehe 4.7 Templates ab Seite 191*
Automatisches Anordnen	Ordnet die Module auf dem Workspace automatisch an. *Siehe 3.3.2 Workspace auf Seite 42*
Ausführen abbrechen	Hält eine laufende Berechnung an. *Siehe 3.6.2 Analyse Abbrechen auf Seite 56*

Tabelle 3.2.: Menü „Bearbeiten“

Menüeinträge „Ansicht“	**Funktion**
Modul–Inspektor einblenden	Öffnet den Modul–Inspektor *Siehe 3.5 Modul–Inspektor auf Seite 51*
Übersichtsfenster einblenden	Blendet das Übersichtsfenster ein und aus. *Siehe 3.3.2 Übersichtsfenster auf Seite 46*
Ansichten	Hier werden alle geöffneten View–Fenster angezeigt, die Sie durch Anwählen des jeweiligen Menüeintrags in den Vordergrund holen können.
Alle Ansichten einrasten	Dockt alle geöffneten View–Fenster in das Hauptfenster. *Siehe 3.3.2 Register auf Seite 46*
Automatisches Einrasten	Aktiviert das automatische Einbinden beim Verschieben der Views ins Hauptfenster. *Siehe 3.3.2 Register auf Seite 46*

Tabelle 3.3.: Menü „Ansicht“

Menüeinträge „Hilfe“	Funktion
Information	Zeigt Informationen über Programm und Version an.
Statusmeldungen	Blendet einen historisierten Statusverlauf des aktuellen Projektes ein.

Tabelle 3.4.: Menü „Hilfe“

3.3.2. Workspace

Ihr Arbeitsbereich („Workspace“) stellt das aktuelle Projekt visuell dar. Mit Drag&Drop lassen sich neue und vorhandene Module beliebig auf der Oberfläche platzieren. Die Funktion *Bearbeiten/Automatisch anordnen* ordnet die Objekte nach einem einfachen Schema anhand ihrer Verknüpfungen vertikal an. Anschließend erscheint der Bestätigungsdialog, mit dem Sie die neue Anordnung akzeptieren oder die vorherige wiederherstellen können.

Klicken Sie auf den Hintergrund des Workspace und halten die linke Maustaste gedrückt, können Sie einen Rahmen aufziehen, mit dem sich mehrere Module markieren lassen, um sie z.B. in einen Container (siehe 4.7 Templates auf Seite 191) umzuwandeln.

Der Workspace umfasst eine vorgegebene Größe, auf dem die Module platziert werden können. Über die Rollbalken am rechten und unteren Bildrand verschieben Sie den dargestellten Ausschnitt. Für eine bessere Übersicht lässt sich das Übersichts–Fenster (siehe 3.3.2, Seite 46) über die Funktion *Übersichtsfenster einblenden* einblenden, das den kompletten Workspace mit seinen Modulen und den derzeitigen Bildausschnitt verkleinert darstellt.

Module

Ein Modul repräsentiert einen Baustein Ihrer Analyse auf dem RayQ–Server. Die Module lassen sich miteinander verbinden, um

einen Datenfluss von einem Modul in ein anderes herzustellen. Es gibt verschiedene Modul–Typen, die in verschiedene Kategorien gruppiert sind. Jeder Kategorie ist eine eigene Farbe zugeordnet, um z.B. Datenbank- und Berechnungs–Module leicht voneinander unterscheiden zu können.

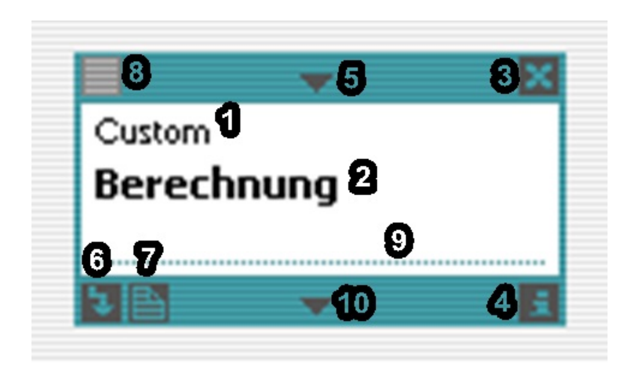


Abbildung 3.5.: Modul

1. Modul–Typ
2. Modul–Name
3. Modul löschen
4. Modul–Inspektor
5. Verbinder, Eingang
6. Execute
7. Memo anzeigen
8. Status–Anzeige
9. Verarbeitungs–Fortschritt
10. Verbinder, Ausgang

Modul–Typ und -Name. In der ersten Zeile wird der Typ des Moduls angegeben, in der zweiten der Name, der vom Benutzer über den Modul–Inspektor beliebig verändert werden kann.

Status und Fortschritt. Das Status–Feld ist anfangs grau gefärbt, da das Modul für Eingaben und Berechnungen bereit ist, aber noch keine gültigen Daten enthält (siehe 2.3.2 Gültigkeit der Daten auf Seite 27). Es ist grün gefärbt, sobald die Daten gültig sind. Rot bedeutet, dass das Modul gesperrt ist, wenn z.B. die Daten aktualisiert werden und eine Änderung durch den Benutzer nicht möglich ist. Führt das Modul gerade eine Berechnung aus, läuft ein gelber Balken kontinuierlich durch das Feld. Eine Änderung der Einstellungen ist zu diesem Zeitpunkt ebenfalls nicht

möglich, jedoch lässt sich die Berechnung durch den Benutzer abbrechen (siehe 3.6.2, Seite 56).

Der Verarbeitungs–Fortschritt wird in einem über die Breite des Moduls gezeichneten Balken (9) dargestellt, während eine Berechnung ausgeführt wird. Erreicht der Balken den rechten Rand, ist die Verarbeitung in dem jeweiligen Modul abgeschlossen. Den Gesamtfortschritt einer Analyse können Sie an der Status–Anzeige (Abbildung 3.4, Seite 38, (8)) des Hauptfensters ablesen.

Ein- und Ausgang. Mit einem kleinen Dreieck in der Mitte des oberen und unteren Randes werden die Verbindungspunkte zu anderen Modulen dargestellt. Klicken Sie auf das untere Symbol, den „Ausgang“, um eine Verbindung zu einem anderen Modul herzustellen. Klicken Sie anschließend auf dessen oberes Symbol, den „Eingang“, wird ein Link hergestellt.

Execute. Startet die Berechnung des jeweiligen Moduls. Der Vorgang kann nur dann ausgeführt werden, wenn gültige Daten an dem Modul anliegen. Ist dies nicht der Fall, wird in gegebenenfalls davor liegenden Modulen ebenfalls die Execute–Prozedur ausgeführt, bevor die Verarbeitung in dem von Ihnen gewählten Modul gestartet wird. Je nach Komplexität der Analyse kann dieser Vorgang einige Zeit in Anspruch nehmen.

Modul löschen. Ein Modul kann gelöscht werden, so lange es mit keinem anderen Modul in Beziehung steht. Entfernen Sie gegebenenfalls alle mit dem Modul verbundenen Links, bevor Sie es löschen.

Tipp!
Statt zweimal zu klicken, können Sie auch die linke Maustaste gedrückt halten und den Link zum nächsten Modul „ziehen“.

Hinweis!
Beim Löschen gehen alle im Modul gespeicherten Daten und Einstellungen verloren.

Links

Die Links stellen die Verbindung zwischen zwei Modulen in einer Richtung her. Dabei werden dem verbundenen Modul alle ausgehenden Tabellen des vorhergehenden zur Verfügung gestellt. Die Links erscheinen zur besseren Unterscheidung in der jeweiligen Farbe des ausgehenden Moduls.

Links löschen. Mit einem Klick mit der rechten Maustaste auf einen Link öffnen Sie das Kontext–Menü, das Ihnen die Funktion zum Löschen des ausgewählten Links anbietet.

Hinweis!
Ein Link wird rot dargestellt, sobald sich der Mauszeiger direkt über ihm befindet.

Hinweis!
Bitte beachten Sie, dass das Löschen eines Links Auswirkungen auf weitere Module in der Kette haben kann und gegebenenfalls Einstellungen und Daten verloren gehen!

Übersichtsfenster

Das Overview–Fenster (Abbildung 3.6) bietet Ihnen einen Überblick über den gesamten Workspace. Klicken Sie mit der Maus in das Übersichtsfenster, können Sie mit gedrückter linker Maustaste den Ausschnitt des Workspace direkt verschieben. Das Übersichtsfenster wird immer im Vordergrund dargestellt, damit es nicht von anderen Fenstern verdeckt wird und immer erreichbar bleibt. Geöffnet wird es über das Hauptmenü *Ansicht/Übersichtsfenster einblenden*.

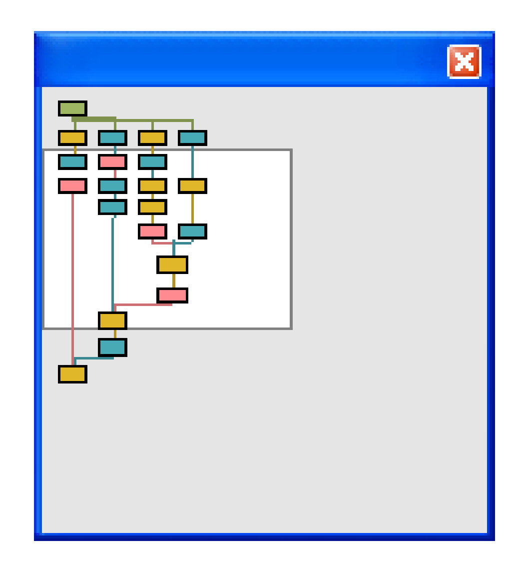


Abbildung 3.6.: Projekt–Übersichtsfenster

Register

Nach dem Öffnen sind alle Übersichts–Fenster (siehe 3.3.2, Seite 46) eigenständige Fenster und können beliebig auf Ihrer Windows–

Oberfläche platziert und skaliert werden. Für eine bessere Übersicht lassen sich die View–Fenster auch direkt in das Hauptfenster einbinden. Verschieben Sie ein Übersichts–Fenster mit der Maus über das Hauptfenster, so „dockt“ es ein. Unterhalb des Workspace befindet sich der Kontrollbereich der Seiten in Form von Register–Karten (Abbildung 3.7), mit denen Sie zwischen den einzelnen Fenstern umschalten können. Möchten Sie eines der Fenster wieder auslösen, klicken Sie mit der Maus doppelt auf die Register–Lasche oder ziehen sie mit der Maus „heraus“. Über das Kontext–Menü mit der rechten Maustaste haben Sie zusätzlich die Möglichkeit, einzelne Fenster direkt zu schließen. Lediglich der Workspace verbleibt immer an seinem angestammten Platz und kann weder geschlossen, noch herausgelöst werden.

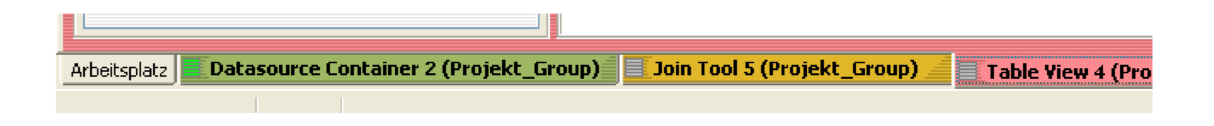


Abbildung 3.7.: Register

Im Menü *Ansicht/Automatisches Einrasten* kann das automatische Einbinden der Fenster beim Verschieben deaktiviert werden. Im Vollbildmodus des Hauptfensters, ist das Einrasten automatisch abgeschaltet.

Tipp!
Halten Sie die Strg–Taste gedrückt, wenn Sie beim Verschieben verhindern möchten, dass ein Fenster gedockt wird.

Tipp!
Über die Funktion *Ansicht/Alle Ansichten einrasten* lassen sich alle offenen View–Fenster in das Hauptfenster einfügen.

Aktive und markierte Module

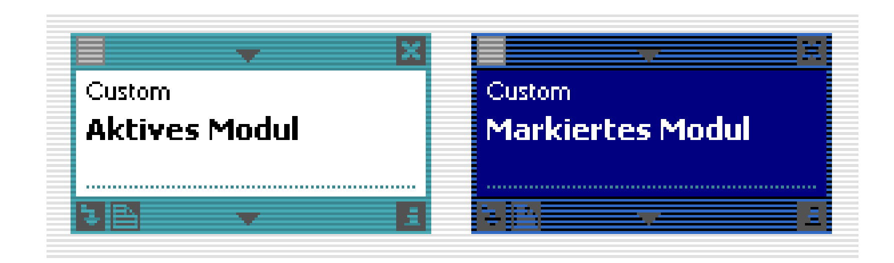


Abbildung 3.8.: Aktives Modul (links), markiertes Modul (rechts)

Das Programm unterscheidet zwischen einem aktiven und mehreren markierten Modulen (Abbildung 3.8).

Das aktive Modul (links) wird mit hellem Hintergrund dargestellt und liefert z.B. die Daten für den Modul–Inspektor (siehe 3.5, Seite 51). Klicken Sie einmal mit der Maus auf ein Modul, um es zu aktivieren.

Klicken und Ziehen Sie mit der Maus auf dem Workspace einen Selektionsrahmen, lassen sich ein oder mehrere Module markieren. Die Module ändern ihre Farbe auf die im System voreingestelle Markierungsfarbe (z.B. dunkelblau bei Windows–Standard–Einstellungen). Markierte Module stehen für Mehrfachfunktionen, wie z.B. das Umwandeln in einen Container (siehe 4.7 Templates auf Seite 191), zur Verfügung.

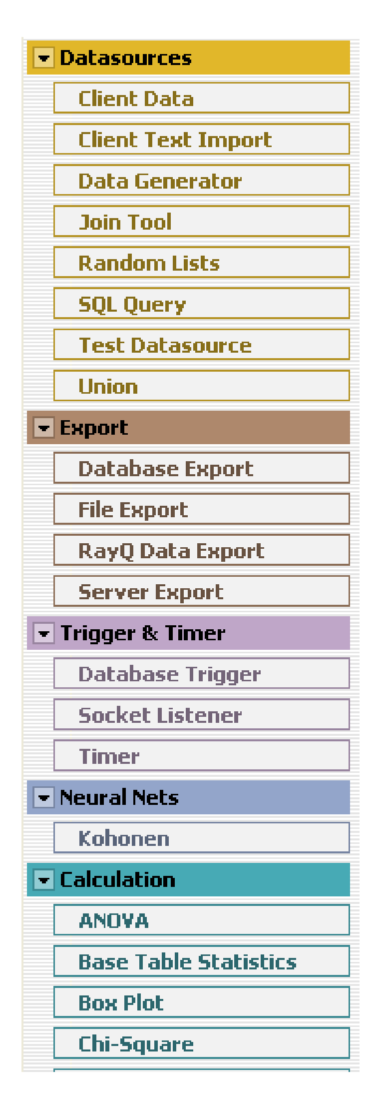


Abbildung 3.9.: Modulleiste

3.4. Modulleiste

In der Modul–Leiste (Abbildung 3.9) sind alle auf dem Server verfügbaren Modul–Typen aufgelistet. An erster Stelle stehen stets die im System verfügbaren Datenbanken.

Der Umfang und die Benennung der Modul–Typen kann variieren und hängt von der bei Ihnen eingesetzten RayQ–Server–Version ab.

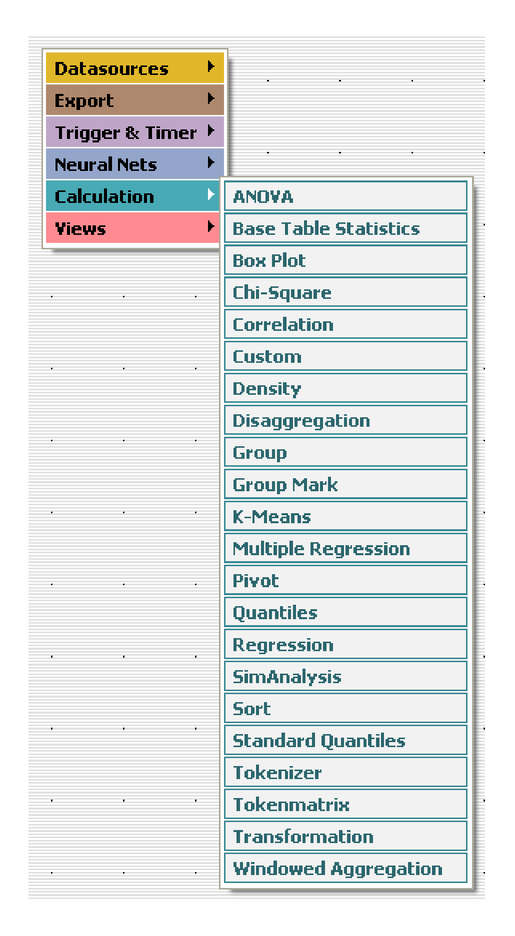


Abbildung 3.10.: Modul–Popup

Mit gedrückter linker Maustaste ziehen Sie ein Modul auf den Workspace und legen es dort an beliebiger Stelle ab.

Hinweis!
Mehr zu den verschiedenen Modul–Typen erfahren Sie unter „RayQ–Module“ ab Seite 63.

Alternativ klicken Sie mit der rechten Maustaste auf den Hintergrund des Workspace, Sie erhalten die Modul–Leiste zusätzlich als Popup–Menü (Abbildung 3.10). Wählen Sie ein gewünschtes Modul aus, wird es an der Stelle auf dem Workspace abgelegt, auf die Sie mit der Maus geklickt hatten.

3.5. Modul–Inspektor

Dem Modul–Inspektor begegnen Sie an verschiedenen Stellen in RayQ. Er ist das zentrale Werkzeug zum Ändern der Eigenschaften eines Moduls. Abhängig von dessen Typ und den vorhandenen Daten zeigt er die jeweiligen Einstellungsmöglichkeiten an, mit denen Sie das Modul parametrisieren können.

Die Art und Weise, wie sich die einzelnen Parameter verändern lassen, hängt von deren jeweiligem Datenformat ab. Zahlenwerte und Zeichenfolgen geben Sie direkt in das Feld im Modul–Inspektor ein, Definitionen von Farbverläufen oder mathematischen Formeln werden über eigene Dialoge verändert.

Zum Eingeben oder Verändern eines dieser Werte klicken Sie mit der Maus in das hervorgehobene Feld oder drücken die Enter–Taste, wenn sich die Eingabemarke (symbolisiert durch ein kleines Dreieck) in der entsprechenden Zeile befindet.

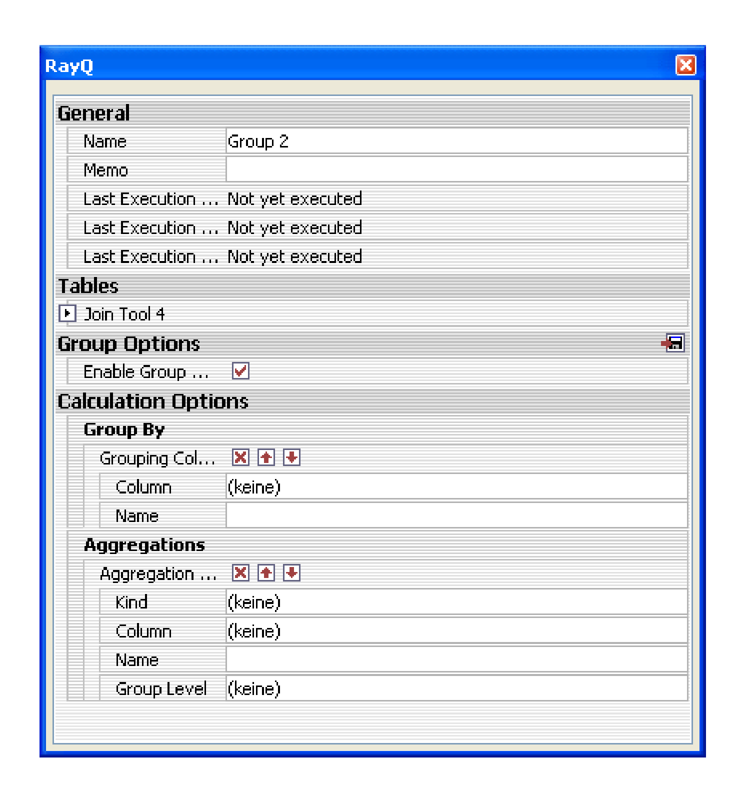


Abbildung 3.11.: Modul–Inspektor

Wichtig!
Eine Änderung wird beim erneuten Drücken von Enter, beim Verlassen des Feldes oder beim Bestätigen des angezeigten Dialogs übernommen.

Hinweis!
Beachten Sie bitte, dass es aufgrund variierender Antwortzeiten im Netzwerk zu einer Verzögerung kommen kann, bis die Änderung vom Server übernommen oder im Fehlerfall abgelehnt wird.

Über das Hauptmenü *Ansicht/Modul–Inspektor einblenden* (siehe Tabelle 3.3) oder das Symbol 4 in Abbildung 3.5 öffnen Sie den Modul–Inspektor, der in einem eigenen Fenster dargestellt wird, das immer im Vordergrund angezeigt wird. Er enthält die Daten des jeweils im Workspace ausgewählten Moduls, das an seinem hellen Hintergrund erkennbar ist.

Jedes View–Fenster (Seite 55) enthält zudem einen Modul – Inspektor, der dem jeweiligen Modul fest zugeordnet ist. So haben Sie mit den View–Fenstern die Konfiguration mehrerer Module parallel im Griff, während der „freie“ Modul–Inspektor des Hauptfensters stets die Informationen des aktuellen Moduls zeigt.

Die Parameter sind bei den meisten Modulen in folgende Gruppen unterteilt:

General. Modulname und Notizen sowie die Information über Start, Ende und Laufzeit des letzten Berechnungslaufs.

Tables. Die Liste aller im Modul verfügbaren Tabellen, sowie deren Spalten und Datentypen. Hier lassen sich die Namen der Tabellen und deren Spalten direkt ändern. Tabellen und deren Spalten, die zur Verarbeitung im Modul ausgewählt sind, werden fett gedruckt dargestellt.

> **Tipp!**
> Klicken Sie auf das Dreieck–Symbol links neben den Tabellen–Namen, um die Liste der Spalten auf- und zuzuklappen.

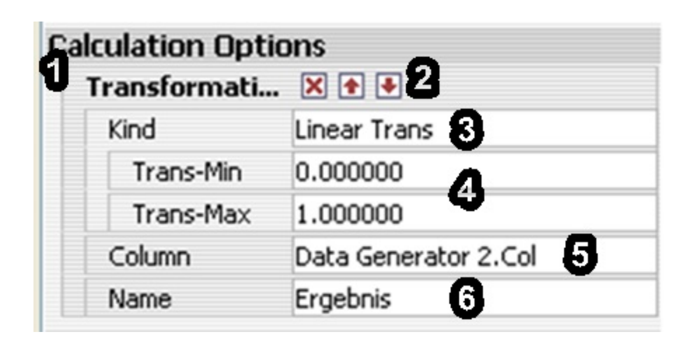


Abbildung 3.12.: Calculation Options

1. Funktions–Block
2. Schaltflächen zum Löschen der jeweiligen Funktion oder zum Einfügen eines leeren Funktionsblocks über oder unter der jeweiligen Zeile
3. Ausgewählte Funktion
4. Funktions–Parameter
5. Spalte, auf die die Funktion angewendet werden soll
6. Name der Ergebnis–Spalte

Module Options. Allgemeine Einstellungen, die das Verhalten eines Moduls steuern.

Calculation Options. In Abbildung 3.12 sehen Sie die Darstel-

lung der von einem Berechnungs–Modul auszuführenden Funktionen mit den jeweiligen Parametern. Es lassen sich über die Schaltflächen (2) je nach Bedarf neue Funktions–Blöcke einfügen. Diese Blöcke werden in der angezeigten Reihenfolge bearbeitet. Abhängig vom jeweiligen Modul–Typ können in der Auswahlbox (3) verschiedene Funktionen ausgewählt werden. Die für die jeweilige Funktion notwendigen Parameter werden unter (4) eingeblendet. Wählen Sie unter (5) die Spalte aus, deren Daten Sie für die Funktionsberechnung verwenden möchten. Der neu erzeugten Ergebnis–Spalte lässt sich unter (6) ein beliebiger Name zuweisen.

Module Parameter. Spezielle Einstellungen, die unter anderem von den ausgewählten Spalten abhängen (z.B. Konfiguration der Graphen im Graphic View)

3.5.1. Memo–Editor

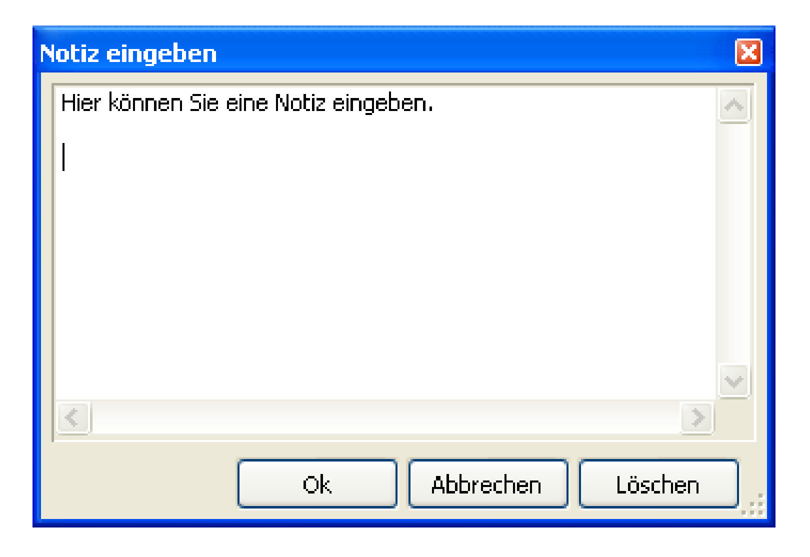


Abbildung 3.13.: Notiz eingeben

Für die Eingabe mehrzeiliger Texte über den Modul–Inspektor ist der Memo–Editor (Abbildung 3.13) zuständig. Der eingegebene Text wird nach Bestätigen mit Ok übernommen, *Löschen* löscht

den gesamten Eintrag.

3.6. View–Fenster

Die View–Fenster bieten eine Detailsicht auf das jeweilige Modul. Sie teilen sich in zwei Bereiche auf: Links ist der Modul–Inspektor (siehe 3.5, Seite 51) eingeblendet, rechts der vom Modultyp abhängige Anzeigebereich.

> **Hinweis!**
> Sie öffnen ein View–Fenster mit einem Doppelklick auf das jeweilige Modul auf dem Workspace.

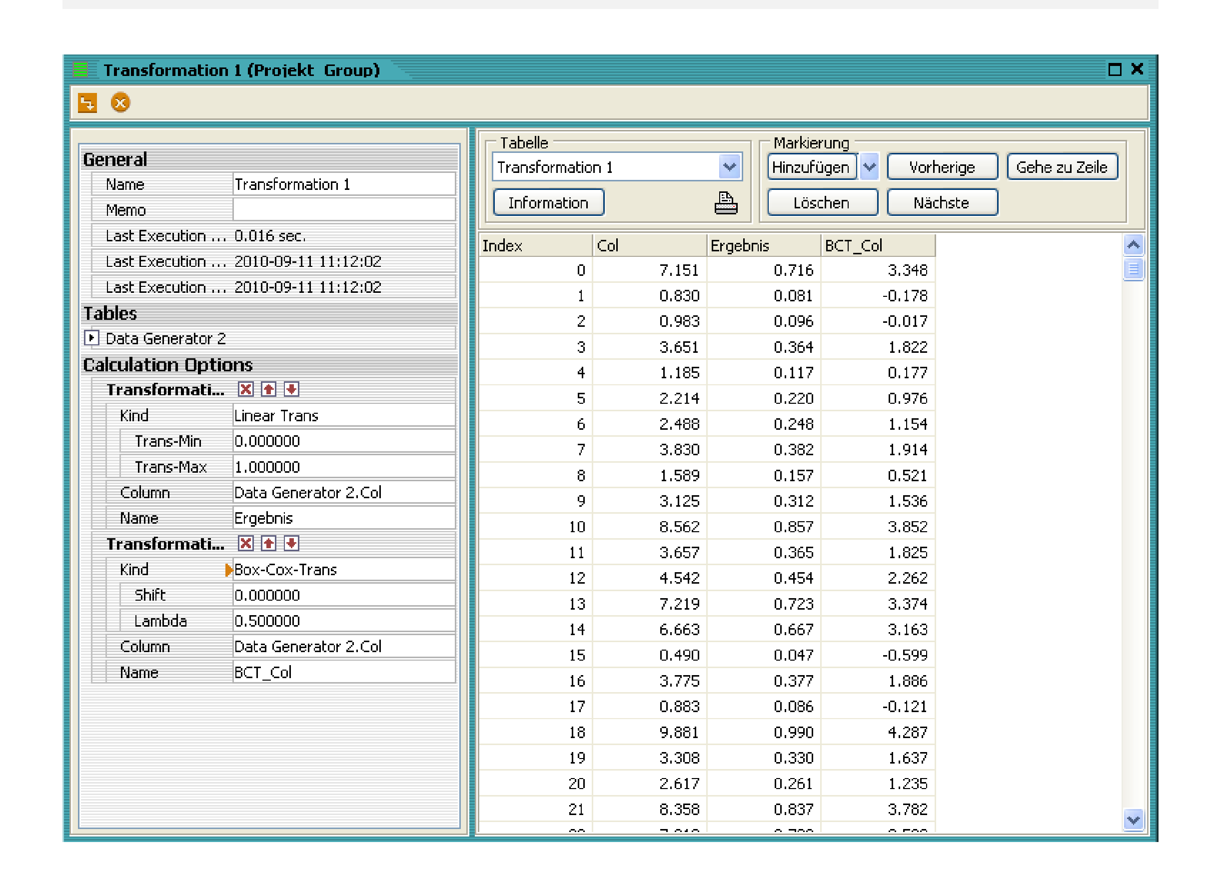


Abbildung 3.14.: Ein View–Fenster: hier ein Berechnungsmodul mit standardmäßig eingeblendeter Tabellenansicht

Module ohne visuellen Teil zeigen in ihren View–Fenstern nur den Modul–Inspektor an. Calculation–Module stellen standardmäßig einen Table–View (siehe 4.8.1, Seite 194) dar, um Ihnen die Konfiguration der Berechnung anhand der sofort sichtbaren Ergebnismenge zu erleichtern.

Eine nähere Beschreibung der individuellen Anzeigebereiche finden Sie ab Seite 63 bei der jeweiligen Modulbeschreibung.

3.6.1. Analyse starten



Abbildung 3.15.: Starten und Beenden einer Analyse

In der Schnellstartleiste (Abbildung 3.15, linkes Symbol) des Hauptfensters sowie in der jedes View–Fensters finden Sie eine Schaltfläche zum Starten einer Analyse. Auch hier wird die Analyse an dem jeweiligen Modul gestartet und gegebenenfalls die Berechnungen an vorhergehenden Modulen ausgeführt.

Für eine bessere Kontrolle der Abläufe zeigen die View–Fenster analog zu den Modulen auf dem Workspace in der Titelleiste ihren aktuellen Zustand über ihre Status–Anzeige (siehe 3.3.2, Seite 42) an.

3.6.2. Analyse abbrechen

In der Schnellstartleiste (Abbildung 3.15, rechtes Symbol) des Hauptfensters sowie in der jedes View–Fensters finden Sie eine Schaltfläche zum Abbrechen einer laufenden Berechnung. Abhängig vom jeweils ausgeführten Modul wird die Bearbeitung sofort oder mit einer kleinen Verzögerung beendet.

Nach dem Abbruch werden nur die von einem Modul teilberechneten Daten verworfen. Die Ergebnisse der Module, die ihre Ausführung vor dem Abbruch beendet haben, bleiben erhalten.

3.7. Projekt schließen

Jedes Hauptfenster des RayQ Designers ist einem Projekt zugeordnet. Schließen Sie das Fenster, wird das jeweilige Projekt geschlossen.

3.7.1. Optionen beim Schließen

Wichtig!
Eine eigene Funktion „speichern“ gibt es nicht — alle Änderungen an einem Projekt oder dessen Daten werden unmittelbar im Cache–System von RayQ gespeichert.

Beim Schließen eines Projekts haben Sie im Dialog aus Abbildung 3.16 die Auswahl zwischen drei verschiedenen Modi:

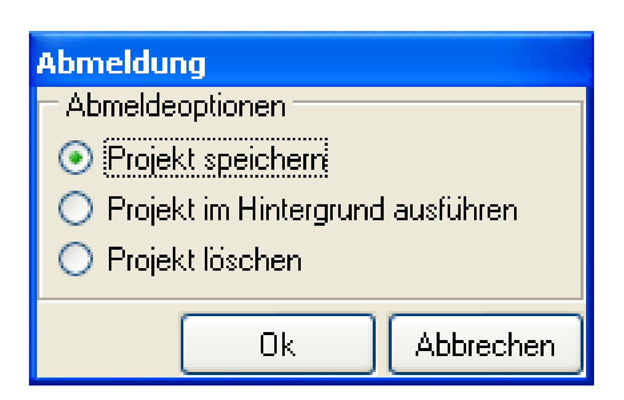


Abbildung 3.16.: Optionen beim Schließen

1. *Projekt speichern* schließt das Projekt und beendet den zugeordneten Server–Prozess. Automatische Prozesse werden

nicht weiter ausgeführt.

2. *Projekt im Hintergrund ausführen* beendet lediglich den Designer, das Projekt bleibt jedoch geöffnet, der zugeordnete Server–Prozess bleibt erhalten. In diesem Modus werden automatische Prozesse, die z.B. durch einen Timer oder Trigger (siehe 4.5, Seite 180) ausgelöst werden, weiterhin ausgeführt.

3. *Projekt löschen* löscht das Projekt und alle gespeicherten Daten aus dem Server–Cashe.

Wichtig!
Die Option *Projekt löschen* kann nicht rückgängig gemacht werden. Alle Einstellungen und Daten eines Projekts gehen verloren.

3.8. Einstellungen speichern

Der RayQ Designer merkt sich Einstellungen, wie z.B. Fenstergröße und -position, benutzer- und projektbezogen. Beim nächsten Öffnen der Anwendung bzw. eines Projekts werden die entsprechenden Einstellungen und Eigenschaften wiederhergestellt.

Hinweis!
Die Einstellungen werden für jeden RayQ Benutzer separat auf dem jeweils verwendeten PC gespeichert, um Probleme z.B. mit unterschiedlichen Bildschirmauflösungen zu vermeiden.

3.8.1. Standardwerte für Module festlegen

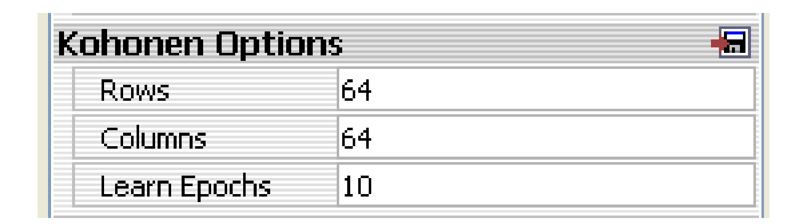


Abbildung 3.17.: Standardwerte speichern

Die *Module Options* (siehe Seite 53) enthalten von Datenquelle und -art unabhängige Einstellungen eines Moduls, wie z.B. Anzahl der Berechnungsschritte oder Ansicht einer Grafik. Möchten Sie abweichend von den Vorgaben eigene Standardwerte definieren, die fortan in jedem neuen Modul dieses Typs verwendet werden, dann klicken Sie mit der Maus auf das Diskettensymbol am rechten Rand (Abbildung 3.17). Die Einstellungen werden projektübergreifend gespeichert, so dass sie auch in neuen oder anderen Projekten bei neu erzeugten Modulen verwendet werden.

3.8.2. Standardwerte zurücksetzen

Möchten Sie die gespeicherten Standardwerte eines Modultyps wieder auf die ursprünglichen Einstellungen zurücksetzen, öffnen Sie das Konfigurationsmenü (Abbildung 3.18) unter *Datei/ Einstellungen*. Wählen Sie den gewünschten Modultyp aus und klicken *Löschen*. Neu erzeugte Module erhalten anschließend wieder die von RayQ vorgegebenen Standardwerte. Möchten Sie die Funktion vorübergehend deaktivieren, ohne die Einstellungen für die einzelnen Module zu löschen, entfernen Sie das Häkchen aus dem Auswahlfeld *Voreinstellungen laden* und bestätigen den Dialog mit *OK*.

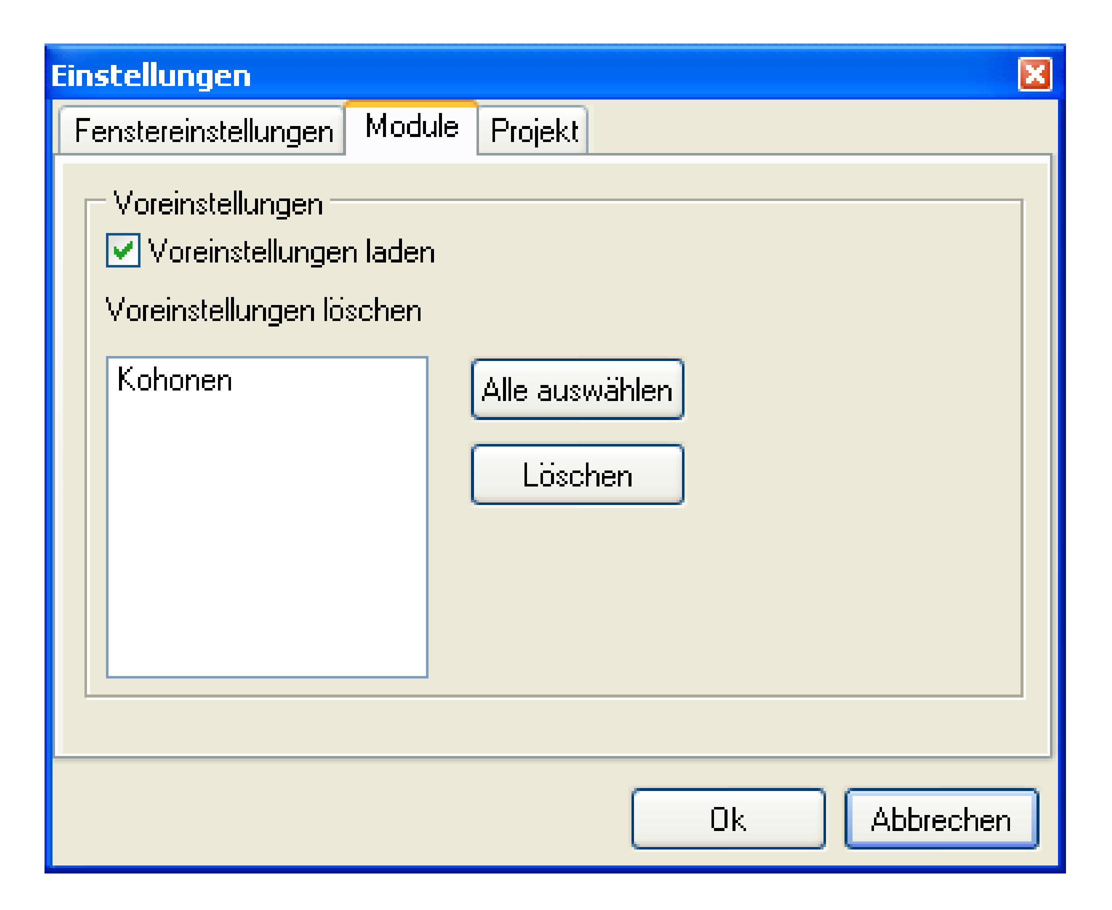


Abbildung 3.18.: Standardwerte zurücksetzen

3.8.3. Fensterposition und Sprache einstellen

RayQ Designer speichert projektübergreifend die Größe und Position der Haupt- und Zusatzfenster (z.B. Übersicht), für jedes Projekt einzeln die Größe, Position der Modulfenster. Beim nächsten Start der Software bzw. Öffnen eines Projekts werden diese Fenstereigenschaften wiederhergestellt.

Im Konfigurationsmenü (Abbildung 3.19) unter *Datei/Einstellungen* lässt sich das Wiederherstellen dieser Eigenschaften einzeln deaktivieren.

Ebenso haben Sie die Möglichkeit die Sprache zu ändern. Eine Wahl zwischen Deutsch und Englisch ist gegeben. Bei Änderung ist ein Neustart erforderlich.

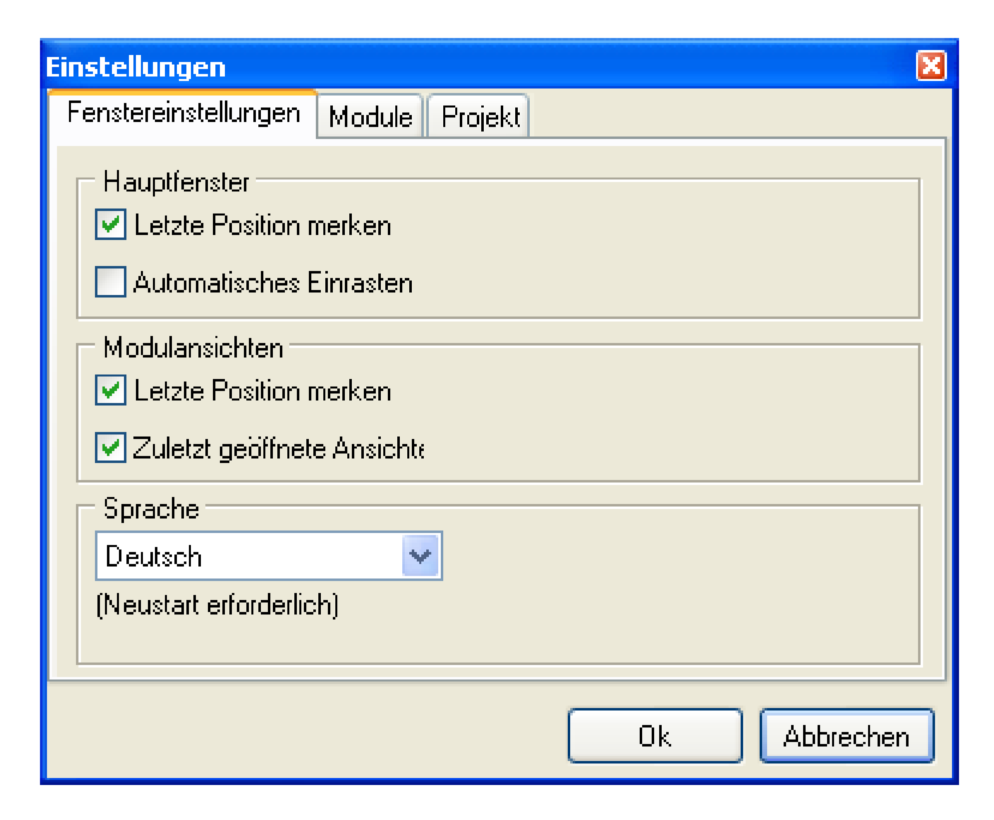


Abbildung 3.19.: Fenstereinstellungen und Sprachwahl

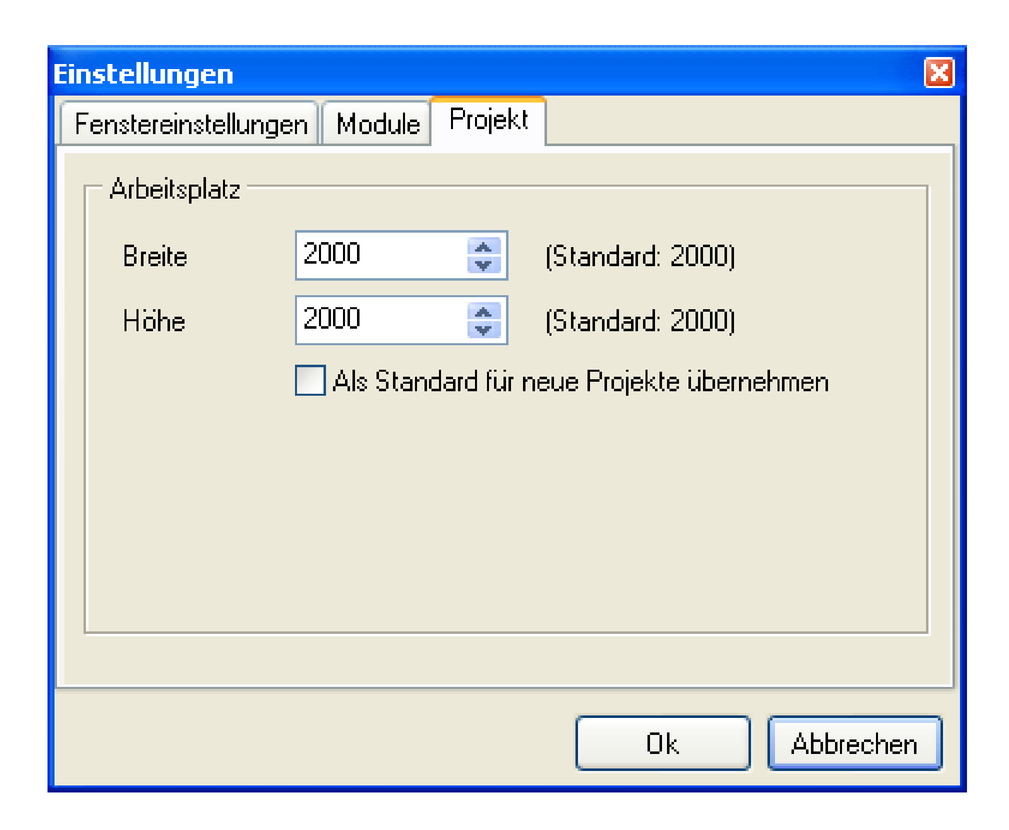


Abbildung 3.20.: Fenstergröße einstellen

3.8.4. Fenstergröße einstellen

Die Fenstergröße lässt sich unter *Datei/Einstellungen* anpassen und für alle Projekte als Standard übernehmen (Abbildung 3.20).

KAPITEL 4

RayQ Module

RayQ–Projekte bestehen aus einzelnen Modulen, die Sie je nach gewünschtem Analyseziel miteinander kombinieren und parametrisieren. Eine Übersicht aller auf Ihrem System verfügbaren Module erhalten Sie im gestarteten RayQ–Designer in der „Modulleiste“ (siehe 3.4, Seite 49).

Die Module sind in logische Gruppen unterteilt, die im RayQ–Designer zudem farblich unterschiedlich gekennzeichnet sind.

Abhängig von Version und Konfiguration des Servers stehen Ihnen die folgenden Module zur Verfügung:

4.1. Databases

In der Kategorie „Databases“ erscheinen alle auf dem Server zur Verfügung stehenden Datenbanken. Die Einträge sind abhängig von der jeweiligen Server–Konfiguration.

Weil Datenbanken in der Regel eine Anmeldung voraussetzen, besitzen alle Database–Module einen Login–Dialog, den Sie mit

dem Symbol (Abbildung 3.5 auf Seite 43, (6)) öffnen. Aus Sicherheitsgründen wird lediglich der Benutzername, nicht aber das Passwort im Modul–Inspektor angezeigt.

4.1.1. Query Designer

Den Query–Designer finden Sie bei allen Datenbank–Modulen, sowie in vereinfachter Form bei dem eigenständigen Join–Modul (siehe 4.2.4, Seite 84). Mit ihm definieren Sie Abfragen, die aus Anweisungen, Bedingungen und „Joins“ bestehen. Für die Abfrage von relationalen Datenbanken (z.B. ORACLE, SQL–Server, Mircosoft Access) wird SQL (Structured Query Language) verwendet. Das Ergebnis einer Abfrage ist eine flache Tabelle mit den entsprechenden Spalten und Zeilen.

Wichtig!
Es muss mindestens eine Abfrage definiert sein, um mit einem Datenbank–Modul gültige Daten zur Verfügung zu stellen.

Die Abfragen lassen sich komfortabel per Maus über die Auswahldialoge oder direkt in einem SQL–Editor zusammenstellen. Gleich welchen Weg Sie wählen, RayQ formuliert Ihre Abfrage als SQL–Anweisung, die Sie jederzeit einsehen und abändern können.

RayQ verwendet dazu den Abfrage–Dialekt der jeweiligen Datenbank. Daraus können sich unterschiedliche SQL–Anweisungen für dieselbe Abfrage. Einige Datenbanken unterstützen bestimmte in RayQ vorgesehene Anweisungen nicht. Dadurch sind entsprechend des Funktionsumfangs des SQL–Dialekts der jeweiligen Datenbank die Abfragemöglichkeiten eingeschränkt.

Umgekehrt haben sie über den SQL–Editor aber auch jederzeit

die Möglichkeit, datenbankspezifische Anweisungen zu verwenden, die der Query–Designer direkt nicht unterstützt.

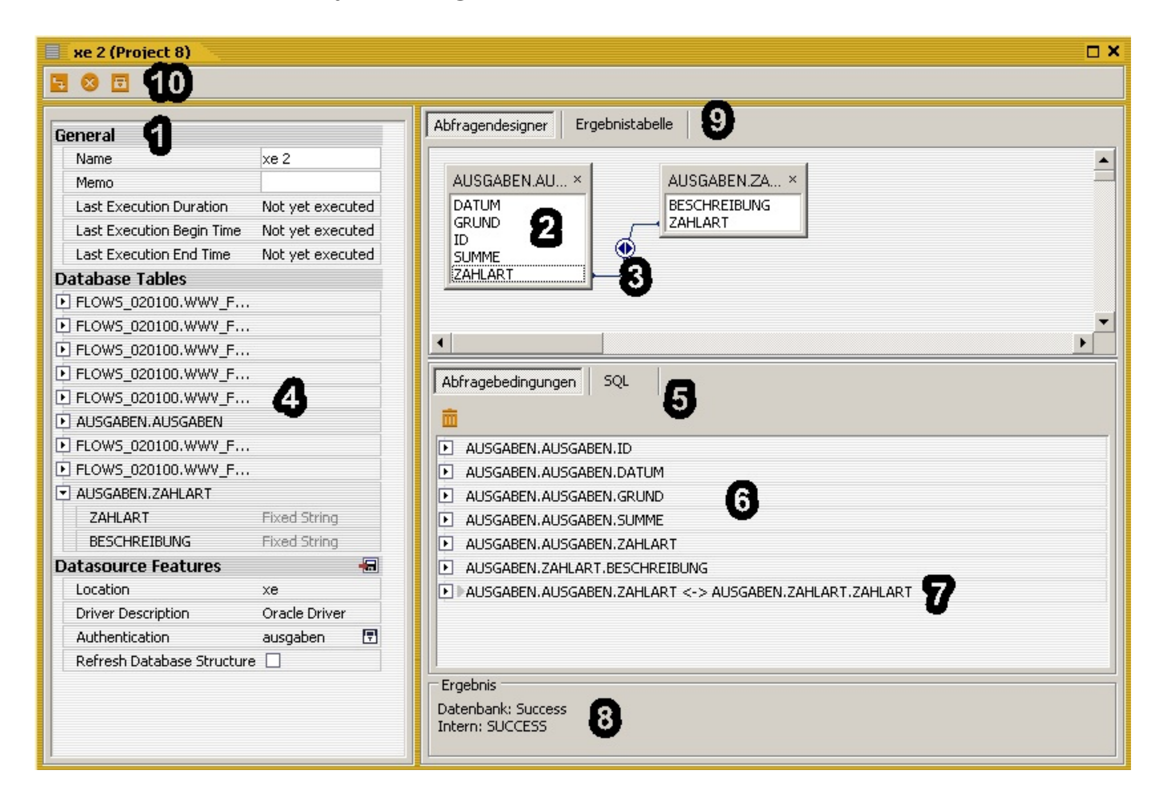


Abbildung 4.1.: Lokaler Daten–Import

1. Modul–Inspektor (siehe Seite 51)
2. Tabelle
3. Join
4. Baumansicht Tabellen und Spalten
5. Umschalten zwischen Conditions- und SQL–Editor
6. Eintrag einer Condition
7. Eintrag eines Joins
8. Ergebnis der SQL–Gültigkeitsüberprüfung
9. Umschalter Query–Designer/Table–View
10. Schnellstart–Leiste

Datenbank–Anmeldung

Für den Zugriff auf eine Datenbank ist in der Regel ein Benutzername und Kennwort notwendig. Klicken Sie auf das Schloss–Symbol (10) in der Schnellstart–Leiste des View–Fensters oder des Moduls auf dem Workspace. Sie werden anschließend zur Eingabe Ihrer Benutzerdaten aufgefordert.

Hinweis!
Die Anmeldedaten der Datenbank sind unabhängig von der Benutzerverwaltung des RayQ–Systems und müssen von Ihrem Datenbank–Administrator separat verwaltet werden.

Hinweis!
Aus Sicherheitsgründen ist nach dem Neuladen eines Projekts eine neue Anmeldung an die Datenbank erforderlich.

Spalten auswählen

Ziehen Sie mit der Maus aus der linken Tabellen–Liste (Abbildung 4.1,(4)) die gewünschte Tabelle oder Spalte in die Conditions–Liste rechts (Abbildung 4.1,(6)): Es wird ein neues SELECT–Statement mit Standard–Parametern angelegt, das Sie anschließend nach Ihren Bedürfnissen abändern können (Erläuterung siehe Tabelle 4.1). Klicken Sie dazu mit der Maus in das betreffende Unterfeld eines Spalteneintrags und nehmen die gewünschten Einstellungen vor.

Bezeichnung	**Werte**
Table Name	Zeigt den Tabellen–Namen der Bedingung an.
Column Name	Wählen Sie die Spalte der Tabelle, für die die Bedingung gültig sein soll. Achtung: Diese Option lässt sich nicht verändern, wenn eine gesamte Tabelle (statt Spaltennamen steht ein „*“ im Auswahlfeld) ausgewählt wurde.
Alias	Vergeben Sie einen alternativen Namen für die Spalte.
Output	Wählen Sie „Yes“, um den Inhalt dieser Spalte mit in das SELECT–Statement aufzunehmen, oder „NO“, um die Spalte nur für eine Bedingung einzusetzen.
Sort Order	Die Sort Order gibt die Sortierreihenfolge der Ergebnisliste an: *none*: keine Sortierung *ascending*: aufsteigend *descending*: absteigend
Function	Aggregat–Funktionen geben auf der Basis anderer Werte einen einzelnen Wert zurück. Wenn Aggregat–Funktionen unter vielen anderen Ausdrücken in der Elementliste einer SELECT–Anweisung verwendet werden, dann muss diese Anweisung für jede ausgegebene Spalte eine GROUP BY–Klausel haben. Dies ist nicht notwendig, wenn die Aggregat–Funktion der einzige von der SELECT–Anweisung erfragte Wert ist. Die unterstützten Aggregat–Funktionen und deren Syntax hängen von der jeweils verwendeten Datenbank ab. Die folgenden Aggregat–Funktionen werden von den meisten Datenbanken unterstützt. *min*: Findet das Minimum der Werte der angegebenen Spalte *max*: Findet das Maximum der Werte der angegebenen Spalte *count*: Zählt alle Zeilen der angegebenen Tabelle oder View.

	avg: Berechnet den Durchschnitt aller Werte der Spalte der angegebenen Tabelle oder View. *variance*: Berechnet die Varianz aller Werte der Spalte der angegebenen Tabelle oder View. *group by*: Gruppiert das Ergebnis der Abfrage nach dieser Spalte.
Condition	Geben Sie eine frei formulierte Bedingung ein, die für die Gültigkeit erfüllt werden muss. Nach Abschluss einer Anweisung wird automatisch eine neue, leere Zeile angefügt, damit Sie weitere Bedingungen eingeben können. Die Bedingungen sind standardmäßig mit „AND“ verknüpft. Klicken Sie mit der Maus auf den Operator, wird die Verknüpfung auf „OR“ umgeändert.

Tabelle 4.1.: Conditions–Parameter bei Spalten

Tipp!
Um schnell zu den gewünschten Daten zu gelangen, ziehen Sie mit der Maus die gewünschte Tabelle (Abbildung 4.1, (4)) nach rechts in die Conditions–Liste (Abbildung 4.1, (6)). Führen Sie das Modul aus, wird die gesamte Tabelle mit allen Spalten und Zeilen eingelesen. Aber bitte beachten Sie, dass es bei großen Datenbanken sinnvoll ist, die Auswahl der in RayQ zu übernehmenden Daten auf die benötigten Spalten und ggf. Zeilen einzuschränken, um die Dauer des Einleseprozesses und die zu übertragende Datenmenge zu optimieren.

Hinweis!
Bitte beachten Sie, dass jede Änderung sowohl vom RayQ–System als auch von der Datenbank eine Gültigkeitsüberprüfung durchläuft. Deshalb kann es zu Verzögerungen bei der Übernahme der vorgenommenen Einstellungen kommen.

Auswahl einschränken

Der Unterbereich *Conditions* einer in der Liste (4.1, (6)) definierten Abfrage erlaubt die Einschränkung der Datenauswahl über SQL–Anweisungen. Tragen Sie in eine *Conditions*–Zeile eine Bedingung ein, wird automatisch eine neue, leere Zeile hinzugefügt, in der Sie weitere Einstellungen vornehmen können. Bleibt sie leer, wird sie bei der Auswertung der Anweisung ignoriert.

Der SQL–Generator stellt aus den angegebenen Bedingungen eine WHERE–Klausel zusammen, die mit den jeweils angegebenen Operatoren AND oder OR verknüpft sind. Klicken Sie mit der Maus auf den am Anfang der Zeile stehenden Operator, um ihn zwischen AND oder OR umzuschalten.

Hinweis!
Bitte beachten Sie, dass während der Eingabe keine Plausibilitätsprüfung der Bedingungen durchgeführt wird. Der SQL–Generator verwendet stets den Spaltennamen als Operanden und Ihre Eingabe als die Fortsetzung des Terms mit Operator und weiterem Operanden.

Join zwischen Tabellen definieren

Im Join–Designer (4.1, (9)) werden alle in der Condition verwendeten Tabellen dargestellt. Weitere Tabellen lassen sich jederzeit mit der Maus durch Klicken und Ziehen aus der Baumansicht (4.1, (4)) hinzufügen.

Zwischen jeweils zwei in den Tabellen aufgeführten Spalten können Sie ebenfalls durch Klicken und Ziehen mit der Maus neue Joins definieren: Wählen Sie in der ersten Tabelle eine Spalte aus und ziehen Sie sie auf die gewünschte Spalte der zweiten Tabelle. Eine Verbindungslinie zwischen diesen beiden Spalten symbolisiert anschließend den Join.

Klicken Sie mit der rechten Maustaste auf das Join–Symbol zwischen den beiden Tabellen, lässt sich der Join–Modus genauer spezifizieren: Standard sind „Inner"–Joins, über das Kontext–Menü können sie in „Left" — oder „Right" — Joins abgeändert oder vollständig gelöscht werden.

Hinweis!
Die Richtung des Symbols bei einem Left- oder Right–Join kann sich in der grafischen Darstellung ändern, z.B. wenn eine Tabelle innerhalb des Designers verschoben wurde. Sie zeigt stets in die logische Richtung. Relevant ist jeweils die Angabe in der Conditions–Liste (4.1, (7)). Die Reihenfolge wird durch das erstmalige Anlegen eines Joins bestimmt: Ziehen Sie die Verbindung mit der Maus z.B. von Tabelle A zu Tabelle B, wird Tabelle A in der Definition immer an erster Stelle stehen, auch wenn sie im Join–Designer rechts neben Tabelle B positioniert ist.

> **Wichtig!**
> Beachten Sie bitte, dass ein Join allein keine Ergebnismenge produziert, sondern lediglich eine vorhandene SELECT–Condition einschränkt. Ist keine SELECT–Anweisung vorhanden, wird die Ergebnismenge leer sein.

SQL–Statement ändern

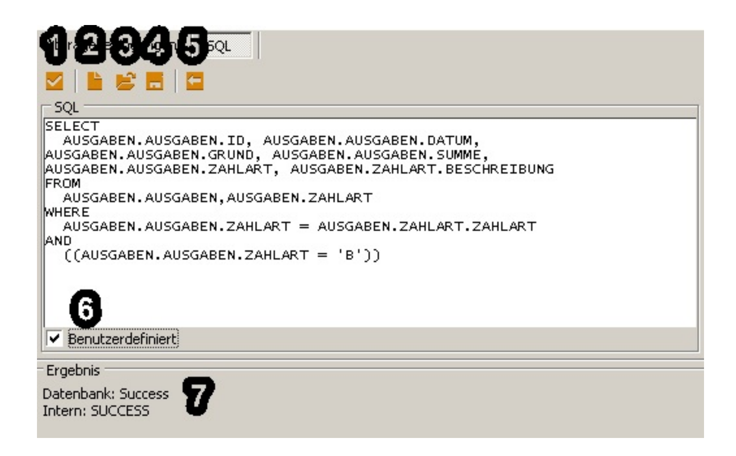


Abbildung 4.2.: SQL–Editor im Query–Designer

1. SQL–Abfrage überprüfen
2. Abfrage löschen
3. Gespeicherte Abfrage laden
4. Aktuelle Abfrage speichern
5. Änderung rückgängig
6. Umschalten zwischen automatischem und manuellem Eingabemodus
7. Ergebnis der SQL–Gültigkeitsüberprüfung

Parallel zur Eingabe über den Join–Designer und der Conditions–Liste wird eine SQL–Abfrage generiert, die Sie durch Umschalten mit der Schaltfläche *SQL* im Editor einsehen können (Abbildung 4.2).

Setzen Sie einen Haken in das Feld *Custom*, ist die manuelle Bearbeitung des SQL–Ausdrucks möglich. Mit einem Mausklick auf das *Confirm*–Symbol oberhalb des Fensters oder durch Drücken von Strg–Enter wird die Änderung an den Server zur Überprüfung gesendet. Das Result–Feld gibt Ihnen darüber Auskunft, ob es sich um einen gültigen Ausdruck handelt. Dabei arbeitet das System zweistufig: Die obere Zeile zeigt das vom jeweiligen Datenbank–Server zurückgelieferte Ergebnis, die untere Zeile das Ergebnis vom RayQ–eigenen SQL–Parser.

Hinweis!
Verwenden Sie Datenbank–spezifische Anweisungen, kann es vorkommen, dass sie vom RayQ–Parser nicht erkannt werden, aber von der Datenbank als gültig bestätigt werden. In diesem Fall können Sie ohne weiteres mit dieser Anweisung arbeiten, jedoch kann sie nur noch über den SQL–Editor und nicht über die Eingabehilfe in der Conditions–Liste und dem Join–Designer verändert werden. Entfernen Sie den Haken im Feld Custom, um wieder in den automatischen Modus zu wechseln. Beachten Sie bitte, dass bei einer Änderung in der Conditions–Liste oder im Join–Designer die von Ihnen manuell bearbeitete SQL–Bedingung durch die im automatischen Teil Definierte überschrieben wird.

Tipp!
Über die „Rückgängig“–Schaltfläche lassen sich bis zu zehn Änderungsschritte Ihrer SQL–Abfrage rückgängig machen. Der Editor merkt sich alle Änderungen, die Sie im grafischen Query–Designer oder direkt im Editor vorgenommen haben.

Speichern und Laden von Abfragen

SQL–Abfragen können Sie über die Menüleiste oberhalb des SQL–Editors als Textdatei abspeichern und laden. Klicken Sie die gewünschte Funktion an und wählen in dem folgenden Verzeichnis–Fenster die gewünschte Datei aus.

Tipp!
Mit der Laden–Funktion können Sie bestehende SQL–Anweisungen aus anderen Anwendungen laden und ausführen.

4.2. Datasources

Unter Datasources befinden sich alle Module, die entweder aus bestehenden Quellen neue Tabellen erzeugen oder eigene Datenreihen für Test– und Kontrollzwecke generieren.

4.2.1. Client Data

Kurzbeschreibung	
Verwendungszweck	Import lokaler Daten
Eingang	-
Ausgabe	Eine Tabelle. Die Spalten und deren Typen entsprechen den importierten Daten.
Besonderheit	Die lokale Datenquelle wird einmalig eingelesen, bei Änderungen an der Quelle müssen die Daten erneut importiert werden.

Der RayQ Designer stellt Ihnen die Möglichkeit zur Verfügung, lokal über den Client erreichbare Datenquellen in ein Analyseprojekt zu importieren.

Der Zugriff auf Datenquellen erfolgt über die die Datenbankschnittstelle „Microsoft ActiveX Data Objects“ (ADO). Welche Datenbanktypen unterstützt werden, hängt von den auf dem System installierten OLE–DB–Treiben ab.

In der Regel steht Ihnen als OLE–DB–Treiber die in Windows–Systemen standardmäßig installierte ODBC–Schnittstelle zur Verfügung, die Zugriff auf die gängigsten Datenbanken ermöglicht.

Wichtig!
Die importierten Daten werden nur durch Aktion des Benutzers in das Analyseprojekt eingespielt. Änderungen an der Datenquelle werden nicht automatisch berücksichtigt. RayQ verwendet ausschließlich die vom Benutzer zuletzt in den Daten–Cache importierten Daten.

Verbindung definieren

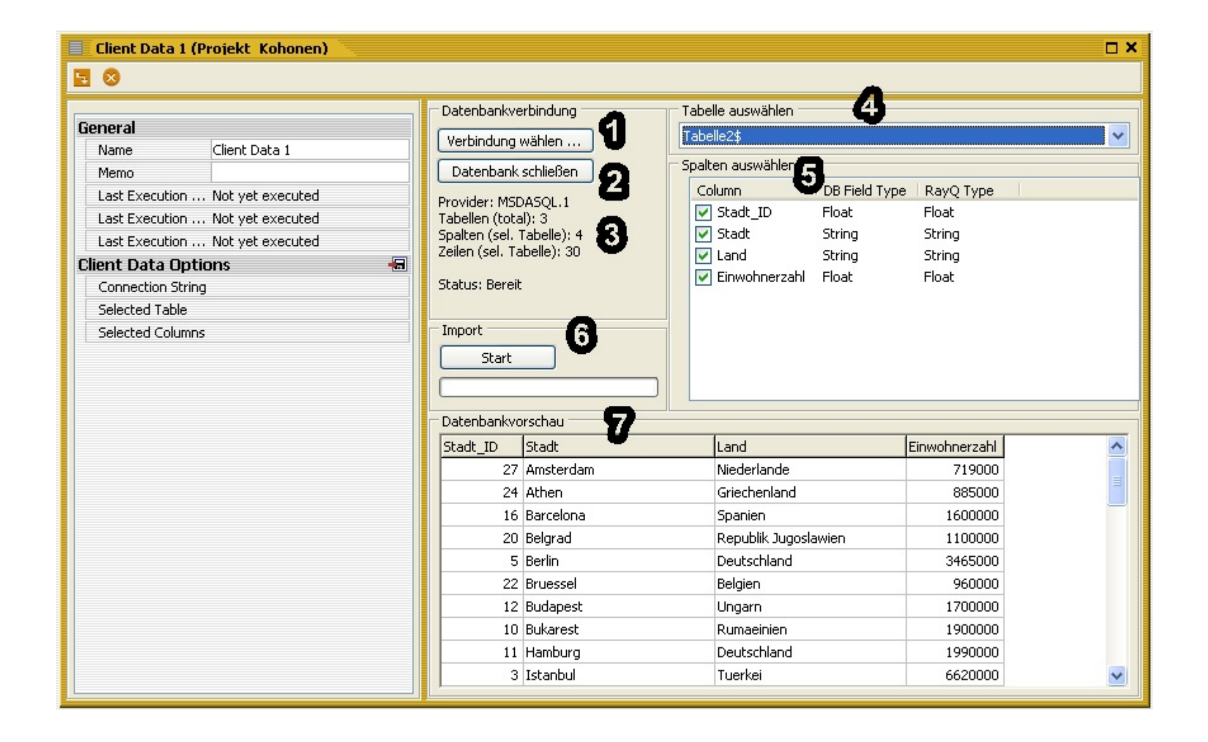


Abbildung 4.3.: Lokaler Daten–Import

1. Verbindungseinstellungen zur Datenquelle
2. Datenquelle öffnen
3. Informationen zur ausgewählten Datenquelle
4. Zum Import ausgewählte Tabelle
5. Zum Import ausgewählte Spalten
6. Import starten
7. Vorschau auf die ausgewählte Datenquelle

Klicken Sie auf die Schaltfläche *Verbindung wählen*, um die Verbindung zu einer Datenquelle anzugeben.

In dem daraufhin angezeigten Dialog (Abbildung 4.4) können Sie direkt den Pfad und Namen einer Excel-, Access- oder Textdatei für den Import angeben.



Abbildung 4.4.: Verbindungseinstellungen, einfache Ansicht

Klicken Sie auf Advanced, um in den erweiterten Einstellungsmodus zu gelangen. Dort können Sie entweder eine Datenverknüpfungsdatei („Datalink Datei"), eine ODBC–DSN oder alternativ direkt eine Verbindungszeichenfolge („Connection String") angeben.

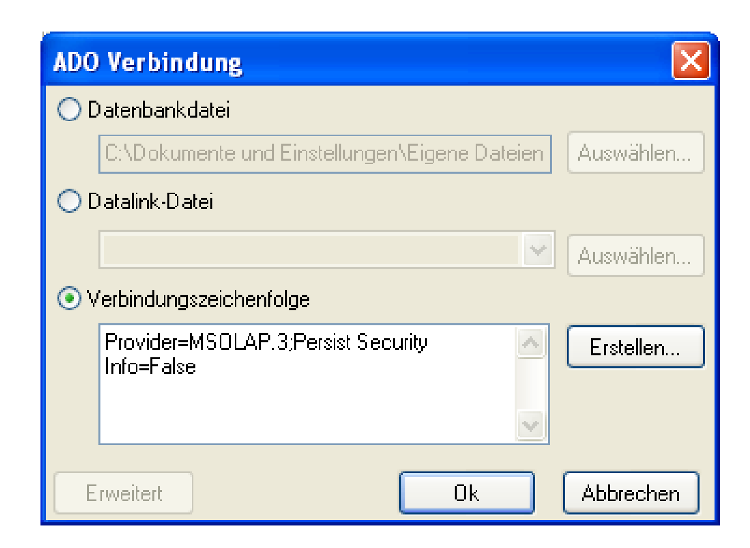


Abbildung 4.5.: Verbindungseinstellungen, erweiterte Ansicht

Connection String

Ein „Connection String" definiert, welcher Datenbanktreiber verwendet werden soll, wie die Datenbank erreicht wird und welche Parameter für den Zugriff notwendig sind. Klicken Sie auf die Schaltfläche Build, um in den von Microsoft ADO bereitgestellten Konfigurationsdialog zu gelangen. Der Inhalt dieses Dialogs variiert und ist vom jeweiligen Datenbank–Treiber abhängig.

Hinweis!
Bitte beachten Sie, dass Datenbank–Kennwörter, die in einen Connection String eingegeben werden, im Klartext lesbar sind.

Tipp!
Klicken Sie auf *Auswählen Datenbankdatei*, um Dateien der Typen Microsoft Excel (XLS), Microsoft Access (MDB) oder Text (ASC, CSV, TAB, TXT) direkt als Datenquelle auszuwählen. Dabei wird ein Connection String mit Standard–Parametern generiert, den Sie direkt verwenden oder mit der Funktion *Erstellen* weiter bearbeiten können.

Verbindung aufbauen

Nachdem Sie eine Datenquelle ausgewählt haben, wird die Verbindung zur Datenbank aufgebaut. Verwenden Sie eine bereits gespeicherten Verbindung, klicken Sie auf *Datenbank öffnen*. In dem unter der Schaltfläche liegenden Informationsbereich erscheinen bei erfolgreicher Verbindung die Konfigurationsdaten der Datenquelle. RayQ fragt beim Öffnen einer Datenquelle standardmäßig nach Benutzernamen und Kennwort. Bitte geben Sie hier die entsprechenden Werte ein oder lassen sie leer, wenn die Datenbank (z.B. beim Import von Textdateien) keine Anmeldedaten benötigt.

Daten importieren

Nach erfolgreicher Verbindung zur Datenquelle müssen Sie die Daten auswählen, die der RayQ Designer importieren soll. Wählen Sie zunächst im Auswahlfeld *Tabelle auswählen* die gewünschte Tabelle der Datenquelle aus. Die Liste *Spalten auswählen* führt

anschließend alle enthaltenen Spalten und deren Datentypen aus. In der dritten Spalte können Sie ablesen, als welcher Datentyp die jeweilige Spalte in RayQ intern weiterverarbeitet wird. Die Tabellenansicht im unteren Bereich der Anzeige gibt Ihnen eine direkte Vorschau auf Ihre Datenquelle.

Wichtig!
Beachten Sie bitte, dass nicht alle Datentypen (z.B. BLOB) verwendet werden können. Weitere Informationen über unterstützte Datentypen entnehmen Sie bitte dem Anhang.

Möchten Sie eine oder mehrere Spalten der ausgewählten Tabelle nicht importieren, so entfernen Sie den Haken am Anfang des jeweiligen Eintrags in der Liste.

Haben Sie alle Auswahlen getätigt, klicken Sie auf Start. Der RayQ Designer beginnt nun, Ihre Datenquelle auszulesen und die Daten an den Server zu senden. Je nach Art und Größe der ausgewählten Daten kann dieser Vorgang mehrere Minuten dauern.

Klicken Sie erneut auf die Schaltfläche, die ihren Text auf Cancel geändert hat, können Sie den Vorgang jederzeit abbrechen.

Nach Abschluss des Vorgangs steht Ihnen über das Client Data Modul eine neue Tabelle mit den importierten Daten zur Verfügung. Ein erneutes Einlesen der Daten ist nicht notwendig, da die Daten im Cache Ihres Analyse–Projekts gespeichert sind. Mit dem Schließen des View–Fensters wird die Verbindung zur Datenquelle abgebaut und wird nicht mehr benötigt. Sie können jedoch jederzeit das Fenster wieder öffnen und die Daten erneut einlesen, z.B. wenn sich der Inhalt zwischenzeitlich geändert hat oder um eine andere Auswahl zu treffen.

4.2.2. Client Text Import

Das Modul *Client Text Import* ermöglicht den Import von Textdateien (*.asc, *.csv, *.tab, *.txt).

Kurzbeschreibung	
Verwendungszweck	Import lokaler Daten
Eingang	-
Ausgabe	Eine Tabelle. Die Spalten und deren Typen entsprechen den importierten Daten.
Besonderheit	Die lokale Datenquelle wird einmalig eingelesen, bei Änderungen an der Quelle müssen die Daten erneut importiert werden.

Nachdem Sie unter *Datei öffnen* die zu importierende Textdatei ausgewählt haben, stehen Ihnen verschiedene Einstellungsmöglichkeiten zur Verfügung.

Bei Zeilengrenzen können Sie zwischen *CRLF* (Carriage Return: Wagenrücklauf, Line Feed: Zeilenvorschub), *CR* (Carriage Return: Wagenrücklauf) oder unter *Benutzerdefiniert:* Ihrer eigenen Einstellung auswählen.

Weiterhin können Zeilen übersprungen werden, indem in das Feld *Zeilen überspringen* eine Ganzzahl eingetragen wird. Enthält die Textdatei Spaltennamen in der ersten Zeilen, so klicken Sie auf *Erste Zeile enthält Spaltennamen*. Ebenso können Sie die Anzahl der Zeilen für die Vorschau einstellen.

Im Bereich der Typkonvertierung lassen sich die Anführungszeichen, die Größe der String–Felder, der Dezimalpunkt bzw. das Date/Time–Format definiert werden.

Neben *Null–Erkennung* kann ein benutzerdefiniertes Zeichen ein-

getragen werden. Alle Einträge mit diesem Zeichen werden im Projekt als *NULL* angezeigt.

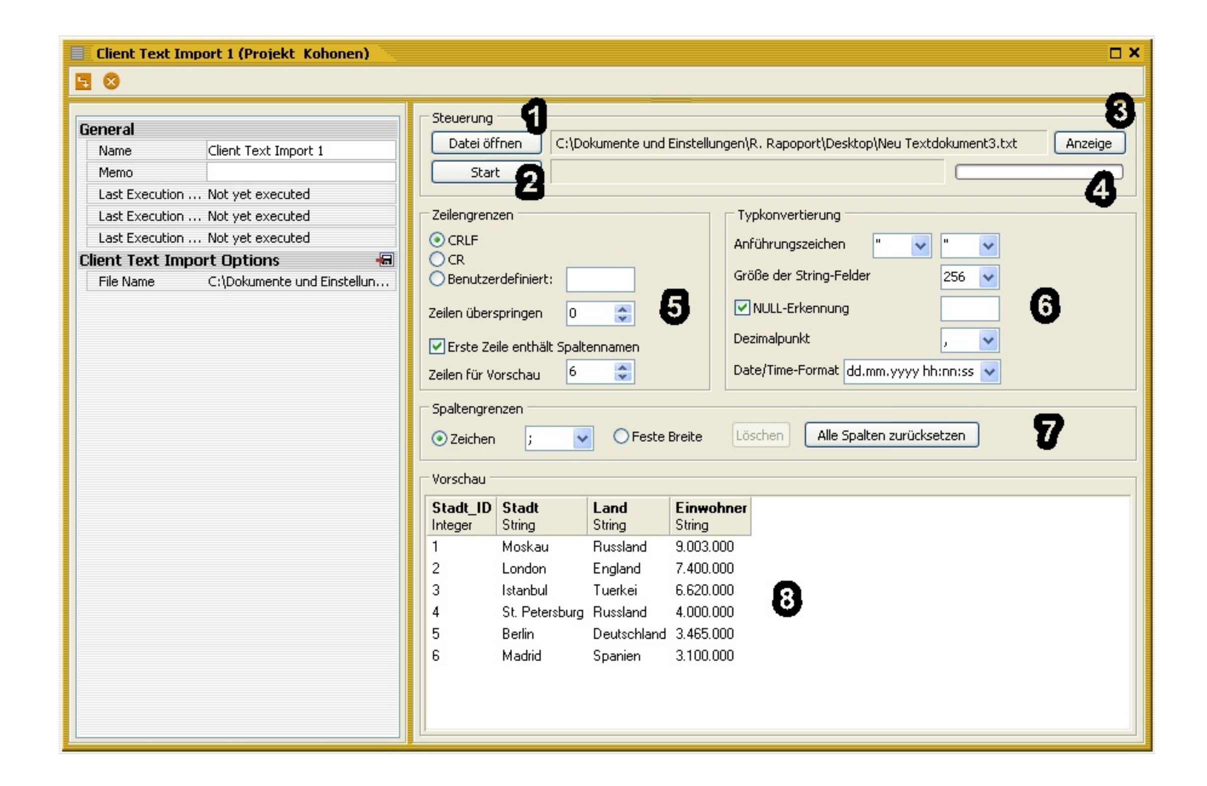


Abbildung 4.6.: Lokaler Daten–Import von Textdateien

1. Verbindung zur Textdatei
2. Importstarten
3. Anzeige der ausgewählten Textdatei im Editor
4. Fortschrittsbalken des Datenimports
5. Einstellungen der Zeilengrenzen
6. Einstellungen zur Typkonvertierung
7. Einstellungen der Spaltengrenzen
8. Vorschau auf die ausgewählte Textdatei

Sind die einzelnen Spalten durch ein Zeichen (”;”, ”,” ,”\t”) getrennt, so wählen Sie die Option *Zeichen* und im nebenstehenden Drop–Down–Feld das trennende Zeichen.

Falls die Spalten der Textdatei durch eine feste Breite getrennt sind, so wählen Sie *Feste Breite*.

Es öffnet sich ein Lineal auf dem Sie durch Klicken chronologisch die Spalten bestimmen können.

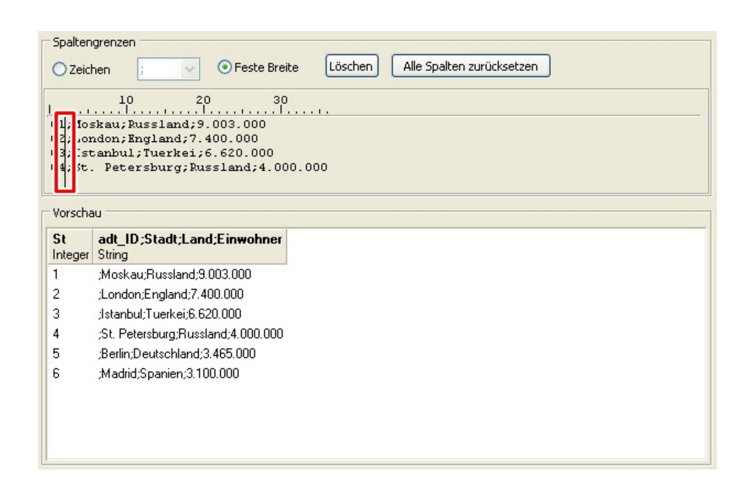


Abbildung 4.7.: Einstellung der festen Breiten der Spaltengrenzen

Im Vorschaufenster kann jede Spalte individuell bearbeitet werden. Mit einem Rechtsklick auf den Spaltennamen können Sie die ausgewählte Spalte deaktivieren, umbenennen, zurücksetzen bzw. dessen Datentypen einstellen.

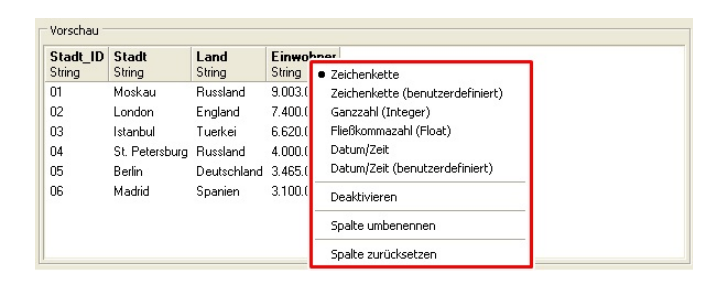


Abbildung 4.8.: individuelle Spalteneinstellungen

Unter der Auswahl *Zeichenkette (benutzerdefiniert)* oder *Zeichenkette (benutzerdefiniert)* kann eine benutzerdefinierte Einstellung vorgenommen werden (siehe Abbildung 4.9).

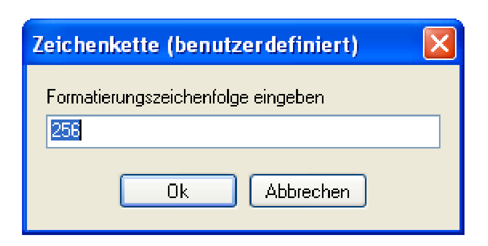


Abbildung 4.9.: Eingabe der Formatierungszeichenfolge

4.2.3. Data Generator

Kurzbeschreibung	
Verwendungszweck	Generieren von zufälligen Daten für Test- und Kontrollzwecke
Eingang	-
Ausgabe	Eine Tabelle mit beliebig vielen Spalten, Typ abhängig von Benutzerauswahl
Besonderheit	Die beim Start einer Analyse erzeugten Zufallsdaten können entweder auf einem einzigen oder jeweils verschiedenen Satz von Zufallsdaten beruhen.

Generiert zufällige oder lineare Zahlenreihen nach Ihrer Vorgabe. Mit *Number of Rows* geben Sie die Anzahl der zu erzeugenden Zeilen an, *Random Initialize* setzt einen Initialisierungswert für den internen Zufallsgenerator. Gleiche Werte erzeugen stets die gleiche Zufallsfolge. Geben Sie „-1“ ein, um den Generator mit der aktuellen Systemzeit zu initialisieren, um bei jedem Ausführen des Moduls eine neue Zufallsreihe zu erzeugen.

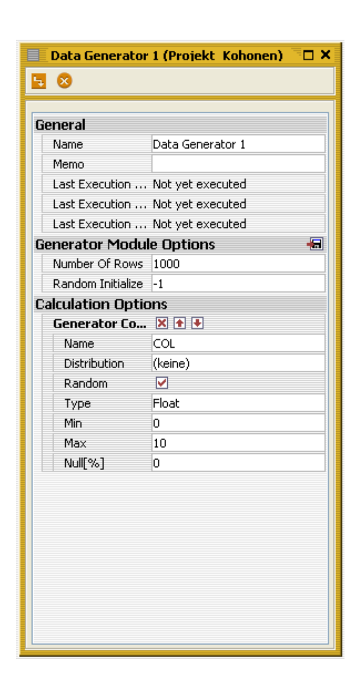


Abbildung 4.10.: Data Generator

Bezeichnung	Werte
Name	Name der zu erzeugenden Spalte
Distribution	Bestimmt die Verteilung der Daten der erzeugten Spalte. *White Noise* erzeugt gleichverteilte, *Normal* normalverteilte Werte.
Random	Legt fest, ob die generierten Daten zufällig oder linear erzeugt werden.
Type	Auszugebender Datentyp: *Float:* Fließkommazahlen *Integer:* Ganzzahlen *String:* Zeichenfolgen *Datetime:* Datumswerte
Min	Unterer Grenzwert
Max	Oberer Grenzwert
Null[%]	Legt den prozentualen Anteil der mit „NULL“ gefüllten Zeilen fest.

Tabelle 4.2.: Parameter Data Generator

4.2.4. Join Tool

Kurzbeschreibung	
Verwendungszweck	Verknüpfen zweier oder mehrerer Tabellen
Eingang	mindestens 2 Spalten aus mindestens 2 verschiedenen Tabellen
Ausgabe	Eine Tabelle, ausgewählte Spalten
Besonderheit	Tabellen beliebiger Herkunft können miteinander verknüpft und mit wählbaren Bedingungen zu einer Datenquelle zusammengefasst werden.

Das Join–Tool verbindet zwei oder mehrere Datenmengen miteinander. Es verwendet den in Datenbanken zur Verfügung stehenden Query–Designer (siehe 4.1.1, Seite 64).

> **Hinweis!**
> Da es sich bei dem Join–Modul um ein RayQ–internes Modul handelt, stehen Ihnen im Query–Designer bestimmte, Datenbank–typische Funktionen nicht zur Verfügung. Die Verwendung der eigentlichen Joins entspricht der im Datasource–Modul.

Auswahl einschränken

Wie im Datasource–Modul gilt, dass im Abschnitt *Abfragebedingungen* aus Spaltenname und der eingegebenen Zeile eine Bedingung analog zur WHERE–Klausel formuliert wird. Verknüpfungen über AND und OR sind ebenfalls möglich, jedoch keine SQL–spezifischen Befehle.
Das Join–Tool bedient sich dem Befehlssatz des Funktionsparsers (Anhang C.1 ab Seite 245) nach folgendem Schema:

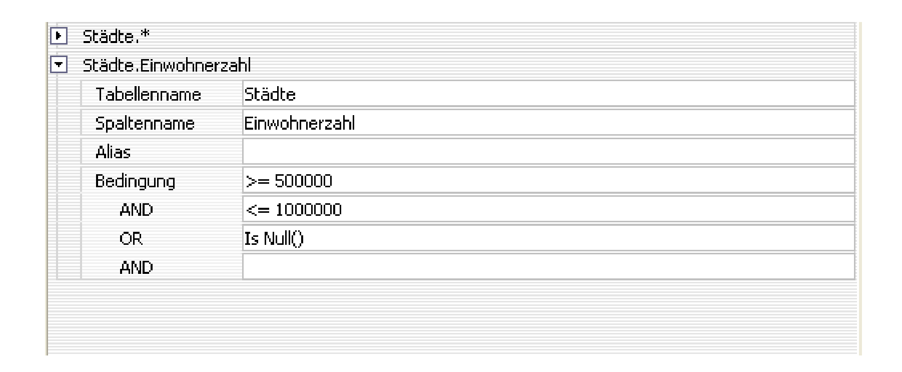


Abbildung 4.11.: Abfragebedingungen im Join Tool

Steht in einer *Abfragebedingungen*–Zeile ein Operator am Anfang, so wird diese Anweisung immer auf die jeweilige Spalte bezogen.

> **Hinweis!**
> Eine Anweisung > 0 wird demnach wie $Spaltenname > 0$ interpretiert.

Steht in einer *Abfragebedingungen*–Zeile ein vollständiger Ausdruck, wird er in dieser Form vollständig ausgewertet. Ein Bezug auf andere Spalten der Tabelle ist möglich, z.B.:

$$SubString\,(Stadt, 0, 1) = "A"$$

Es werden alle Städte mit dem Anfangsbuchstaben „A“ ausgewählt.

Zusätzlich zu den im Funktionsparser verfügbaren Operatoren und Funktionen steht Ihnen der aus SQL bekannte Ausdruck *Is Null()* zur Verfügung, um die aktuelle Zeile der Spalte auf einen NULL–Wert zu überprüfen.

Beispiel

Betrachten wir die folgenden beiden Tabellen (Abbildungen 4.12 und 4.13):

Stadt_ID	Stadt	Land	Einwohnerzahl
27	Amsterdam	Niederlande	719000
24	Athen	Griechenland	885000
16	Barcelona	Spanien	1600000
20	Belgrad	Republik Jugoslawien	1100000
5	Berlin	Deutschland	3465000
22	Bruessel	Belgien	960000
12	Budapest	Ungarn	1700000
10	Bukarest	Rumaeinien	1900000
11	Hamburg	Deutschland	1990000
3	Istanbul	Tuerkei	6620000
8	Kiew	Ukraine	2616000
30	Kischinew	Moldawien	676000
25	Lissabon	Portugal	830000
2	London	Vereinigtes Koenigreich	7400000
6	Madrid	Spanien	3100000
26	Marseille	Frankreich	800000
14	Minsk	Weissrussland (Belarus)	1633000
1	Moskau	Russland	9003000
17	Muenchen	Deutschland	1256000
9	Paris	Frankreich	2152000
18	Prag	Tschechische Republik	1214000
23	Riga	Lettland	897000
7	Rom	Italien	2693000
21	Rotterdam	Niederlande	1040000
19	Sofia	Bulgarien	1110000
4	St. Petersburg	Russland	4000000
29	Stockholm	Schweden	693000
13	Warschau	Polen	1653000
15	Wien	Oesterreich	1600000
28	Zagreb	Kroatien	707000

Abbildung 4.12.: Beispieltabelle I: Städte

Stadt_ID	Fläche	Bevölkerungsdichte
1	1081	9772
2	1572	4784
5	892	3860
6	607	5190
7	1285	2106
8	839	3321
9	105	20774
10	228	8510
11	755	2348
12	525	3231
13	518	3284
15	415	4110
16	100	16056
18	496	2392
19	1039	1158
21	304	1911
23	307	2370
24	39	18713
26	241	3356
27	130	5725
29	188	4164
30	120	4938

Abbildung 4.13.: Beispieltabelle II: Städtinformationen

Aufgabe
An die Beispieltabelle I: *Städte* (Abbilding 4.12) sollen die Felder Fläche und Bevölkerungsdichte aus der Beispieltabelle II: *Stadtinformationen* (Abbilding 4.13) verknüpft werden. Über den Schlüssel *Stadt_ID* lässt sich der Join realisieren.

In Abbildung 4.14 wird ein *OUTER–Join* verwendet. Zeilen aus Beispieltabelle I: *Städte* bleiben auch dann erhalten, wenn in der Beispieltabelle II: *Stadtinformation* die zugehörige *Stadt_ID* nicht existiert. Die Spalten *Fläche* und *Bevölkerungsdichte* erhalten in solchen Fällen der Wert *NULL*.

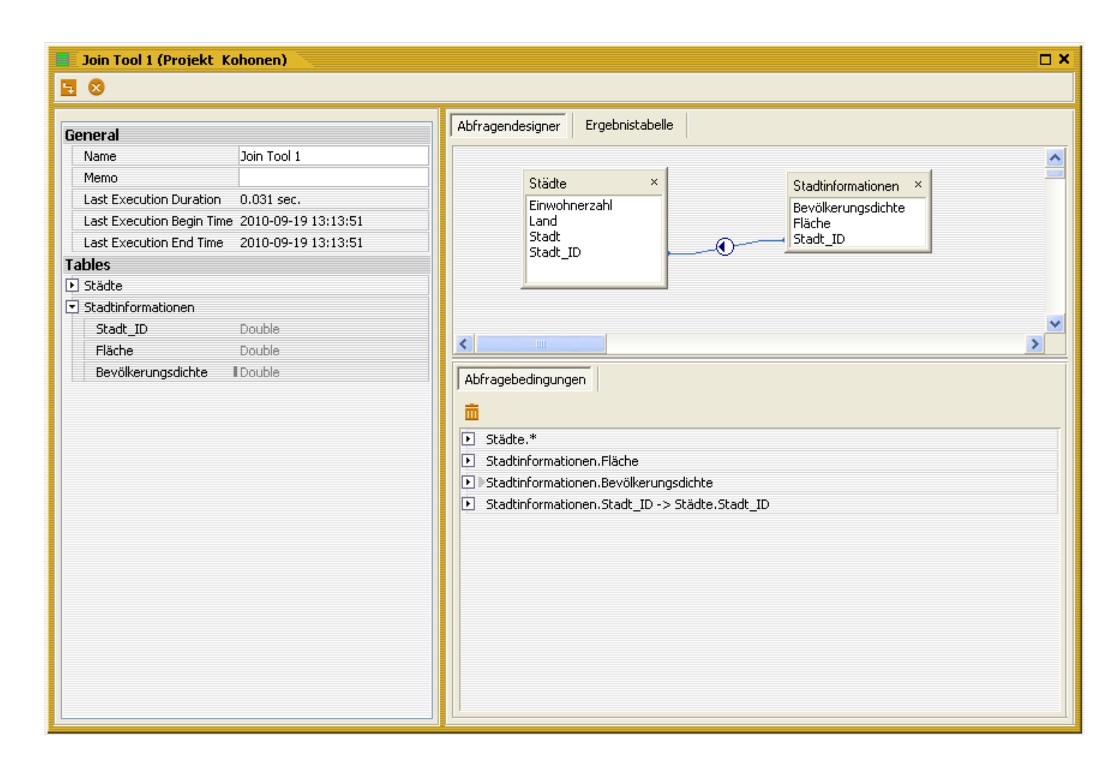


Abbildung 4.14.: Join–Einstellungen

Stadt_ID	Stadt	Land	Einwohnerzahl	Fläche	Bevölkerungsdichte
27	Amsterdam	Niederlande	719000	130	5725
24	Athen	Griechenland	885000	39	18713
16	Barcelona	Spanien	1600000	100	16056
20	Belgrad	Republik Jug...	1100000	-	-
5	Berlin	Deutschland	3465000	892	3860
22	Bruessel	Belgien	960000	-	-
12	Budapest	Ungarn	1700000	525	3231
10	Bukarest	Rumaeinien	1900000	228	8510
11	Hamburg	Deutschland	1990000	755	2348
3	Istanbul	Tuerkei	6620000	-	-
8	Kiew	Ukraine	2616000	839	3321
30	Kischinew	Moldawien	676000	120	4938
25	Lissabon	Portugal	830000	-	-
2	London	Vereinigtes ...	7400000	1572	4784
6	Madrid	Spanien	3100000	607	5190
26	Marseille	Frankreich	800000	241	3356
14	Minsk	Weissrussla...	1633000	-	-
1	Moskau	Russland	9003000	1081	9772
17	Muenchen	Deutschland	1256000	-	-
9	Paris	Frankreich	2152000	105	20774
18	Prag	Tschechisch...	1214000	496	2392
23	Riga	Lettland	897000	307	2370
7	Rom	Italien	2693000	1285	2106
21	Rotterdam	Niederlande	1040000	304	1911
19	Sofia	Bulgarien	1110000	1039	1158
4	St. Petersbu...	Russland	4000000	-	-
29	Stockholm	Schweden	693000	188	4164
13	Warschau	Polen	1653000	518	3284
15	Wien	Oesterreich	1600000	415	4110
28	Zagreb	Kroatien	707000	-	-

Abbildung 4.15.: Ergebnis eines *OUTER–Joins*

Hinweis!
Verknüpft man zur Tabelle A Spalten von Tabelle B mit Hilfe eines *OUTER–Joins*, so erhält die Ergebnistabelle mindestens genau so viele Datensätze wie Tabelle A.

Wie verändert sich das Ergebnis, wenn ein *INNER–Join* verwendet wird?

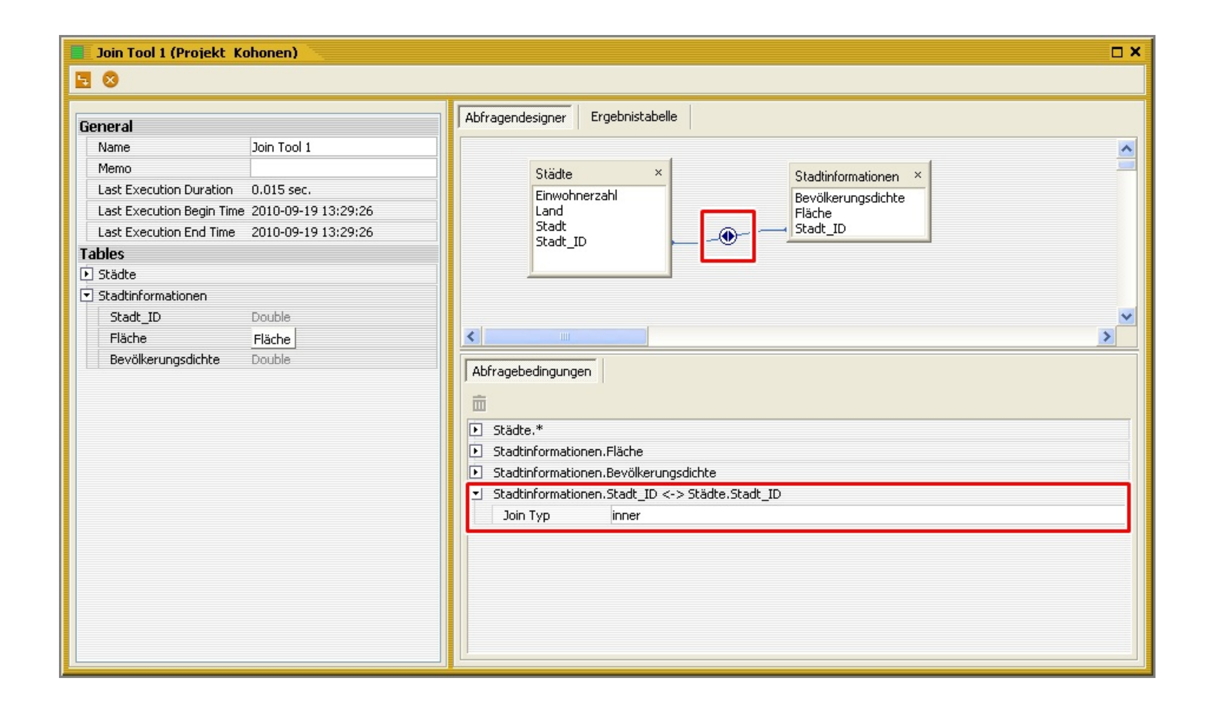


Abbildung 4.16.: Einstellungen eines *INNER–Joins*

Sie haben entweder die Möglichkeit im Abfragedesigner mit Rectsklick auf die Join–Verbindung zu klicken oder unter Abfragebedingen bei *Join Typ* Ihre gewünschten Einstellungen vorzunehmen.

Stadt_ID	Stadt	Land	Einwohnerzahl	Fläche	Bevölkerungsdichte
1	Moskau	Russland	9003000	1081	9772
2	London	Vereinigtes ...	7400000	1572	4784
5	Berlin	Deutschland	3465000	892	3860
6	Madrid	Spanien	3100000	607	5190
7	Rom	Italien	2693000	1285	2106
8	Kiew	Ukraine	2616000	839	3321
9	Paris	Frankreich	2152000	105	20774
10	Bukarest	Rumaeinien	1900000	228	8510
11	Hamburg	Deutschland	1990000	755	2348
12	Budapest	Ungarn	1700000	525	3231
13	Warschau	Polen	1653000	518	3284
15	Wien	Oesterreich	1600000	415	4110
16	Barcelona	Spanien	1600000	100	16056
18	Prag	Tschechisch...	1214000	496	2392
19	Sofia	Bulgarien	1110000	1039	1158
21	Rotterdam	Niederlande	1040000	304	1911
23	Riga	Lettland	897000	307	2370
24	Athen	Griechenland	885000	39	18713
26	Marseille	Frankreich	800000	241	3356
27	Amsterdam	Niederlande	719000	130	5725
29	Stockholm	Schweden	693000	188	4164
30	Kischinew	Moldawien	676000	120	4938

Abbildung 4.17.: Ergebnis des *INNER–Joins*

Hinweis!
In Abbildung 4.17 wird ein *INNER–Join* verwendet. Nur diejenigen Datensätze, dessen *Stadt_ID* in beiden Beispieltabellen vorhanden ist, werden in der Ergebnistabelle ausgegeben.

4.2.5. Random Lists

Im Modul *Random Lists* werden Zufallslisten generiert. Unter *Number Of Lists* wird die Anzahl der Listen bzw. unter *Number Of Values Per List* die Anzahl der Werte pro Liste eingestellt. Der Wertebereich ist durch *Min* und *Max* eingeschränkt.

Kurzbeschreibung	
Verwendungszweck	Stellt Zufallslisten zur Verfügung
Eingang	-
Ausgabe	Eine Tabelle, drei Spalten: 3x Integer
Besonderheit	Das Modul dient nur zu Testzwecken, die Daten lassen sich nicht ändern.

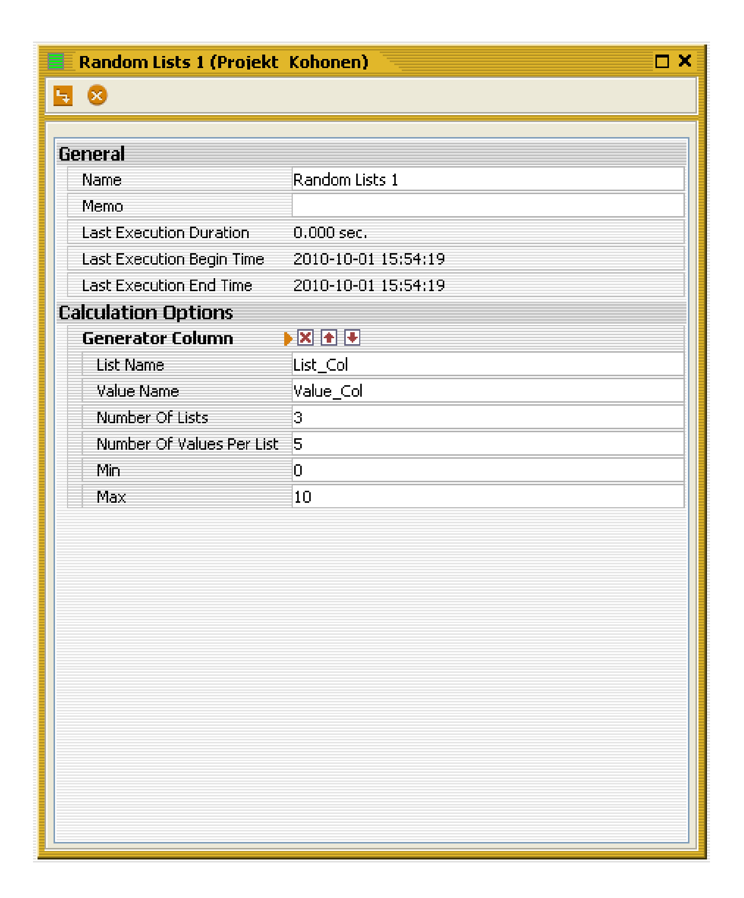


Abbildung 4.18.: Random Lists

Beispiel

Es sollen drei Listen je fünf Werte, die zwischen 0 und 10 liegen, erzeugt werden. Die dazugehörigen Moduleinstellungen können der Abbildung 4.18 entnommen werden. Das Ergebnis sieht wie folgt aus (Abbildung 4.19):

Index	List_Col	Value_Col
0	0	8
1	0	0
2	0	2
3	0	10
4	0	1
5	1	10
6	1	8
7	1	3
8	1	7
9	1	1
10	2	5
11	2	4
12	2	0
13	2	7
14	2	2

Abbildung 4.19.: Ergebnis der Einstellungen aus der Abbildung 4.18

1. erste Liste (List_Col = 0)
2. zweite Liste (List_Col = 1)
3. dritte Liste (List_Col = 2)

4.2.6. SQL–Query

Kurzbeschreibung	
Verwendungszweck	Einfacher SQL Editor zum Abfragen von Daten aus den eingerichteten Datenbanken
Eingang	-
Ausgabe	Eine Tabelle, die Spalten und deren Typen entsprechen den Spalten des Select statements
Besonderheit	-

Das Modul dient dazu eine SQL Abfrage an eine eingerichtete Datenbank zu schicken (Abbildung 4.20), dazu stellt es nur die Möglichkeiten eine Datenbank auszuwählen, sich auf dieser einzuloggen und ein SQL–Statement einzugeben, bzw. zu laden oder zu speichern zur Verfügung (Abbildung 4.21). Es bietet nicht den Komfort der oben genannten grafischen Datenbankmodule, kommt, aber auch ohne deren Overhead aus. So werden zum Beispiel hier nicht die vorhandenen Tabellen ausgelesen und angezeigt, es eignet sich also auch für Situationen, in denen eine größere Datenbank nur langsam oder mit geringer Bandbreite angebunden ist, und gezielt wenige bekannte Daten abgefragt werden sollen.

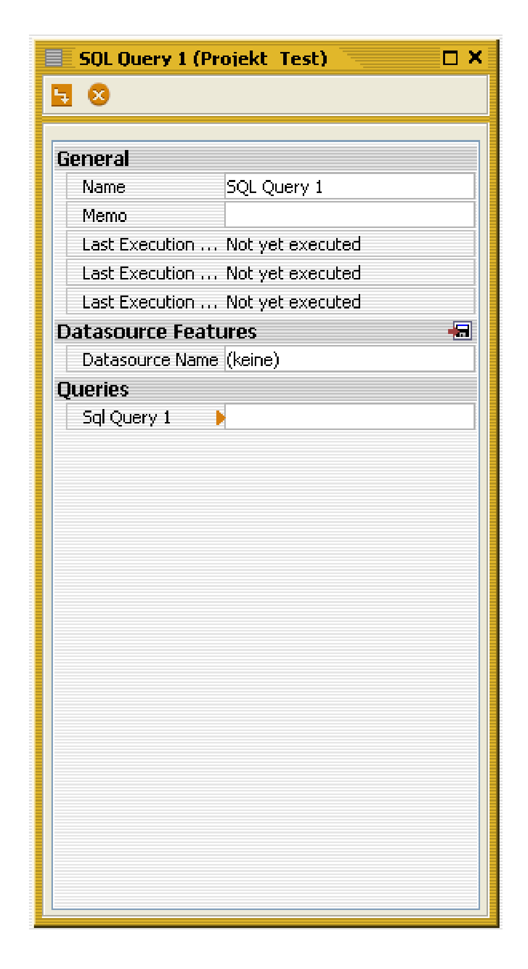


Abbildung 4.20.: Modul: SQL Query

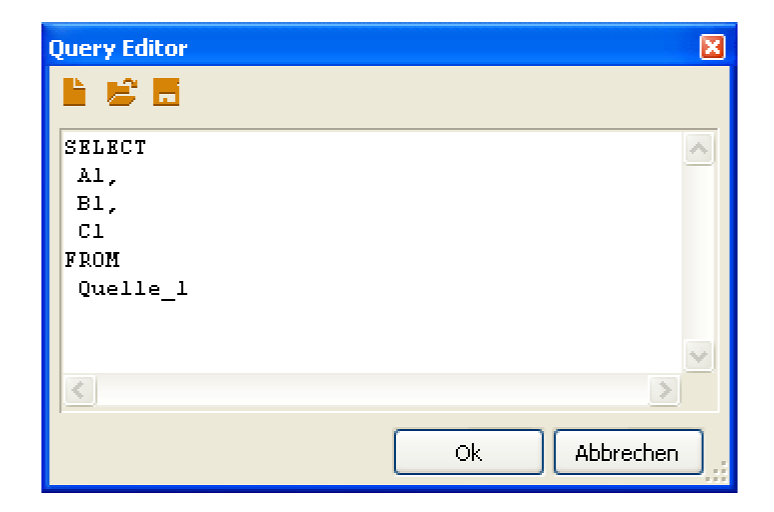


Abbildung 4.21.: Editor zur Eingabe des SQL–Statements

4.2.7. Test Datasource

Kurzbeschreibung	
Verwendungszweck	Stellt Testdaten zur Verfügung
Eingang	-
Ausgabe	Eine Tabelle, vier Spalten: 1x Integer, 3x Float
Besonderheit	Das Modul dient nur zu Testzwecken, die Daten lassen sich nicht ändern.

Das Modul ist für Test- und Übungszwecke implementiert. Es liefert eine fest vorgegebene Datenmenge, bestehend aus einer Index–Spalte und drei Spalten mit Fließkommawerten, die sich in einem dreidimensionalen Raum in sechs gegenüberliegenden Wolken gruppieren.

Die Test Datasource eignet sich z.B. für einfache Versuche mit dem Kohonen–Modul (siehe 4.4.1, Seite 171).

4.2.8. Union

Kurzbeschreibung	
Verwendungszweck	Fügt Daten zweier oder mehrerer Tabellen aneinander
Eingang	beliebig viele Tabellen
Ausgabe	Eine Tabelle, ausgewählte Spalten
Besonderheit	Nur Spalten gleichen Typs können miteinander verbunden werden.

Das Modul führt Spalten unterschiedlicher Datenquellen zusammen. Dabei müssen von jeder eingehenden Datenquelle die gleiche Anzahl Spalten mit denselben Datentypen ausgewählt wer-

den.

In Abbildung 4.22 sehen Sie die Auswahl von jeweils zwei Spalten aus drei verschiedenen Datenquellen. Als Ergebnis erhalten Sie eine Tabelle mit zwei Spalten. Die erste enthält die Verbindung aus den Spalten A1, B2 und C1, die zweite aus A3, B3 und C2. Die Spaltennamen werden immer der ersten Quelle entnommen: A1 und B1.

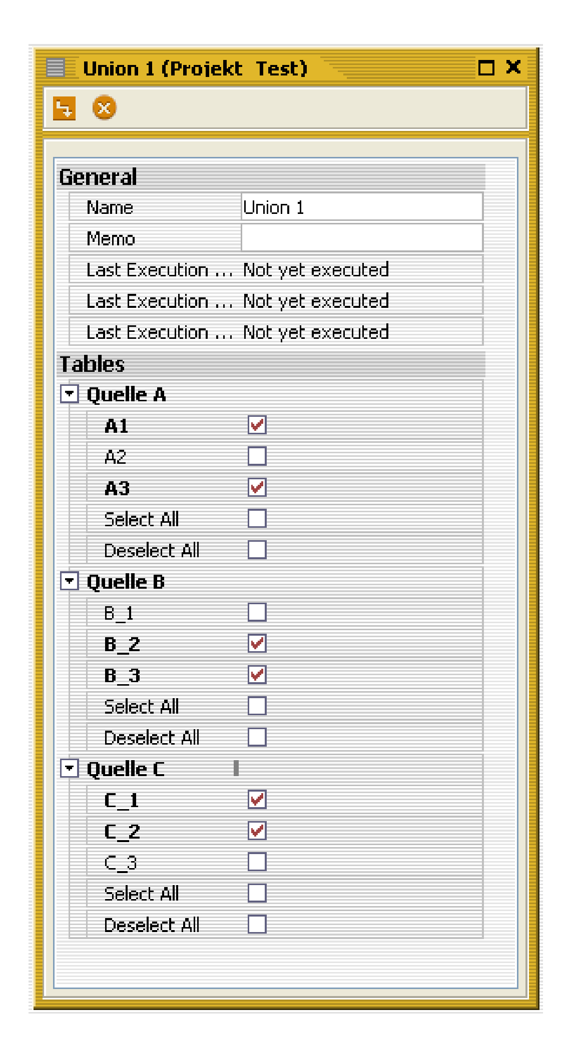


Abbildung 4.22.: Auswahl der zu verbindenden Spalten im Modul Union

Wichtig!
Bei der Auswahl mehrerer Spalten ist die Reihenfolge innerhalb der verschiedenen Datenquellen ausschlaggebend. Verwenden Sie gegebenenfalls mehrere einzelne Union–Module, wenn sich die gewünschte Verbindung aufgrund unterschiedlicher Reihenfolgen nicht in einem einzigen Modul abbilden lässt.

4.3. Calculation

4.3.1. ANOVA

Kurzbeschreibung	
Verwendungszweck	Berechnung von Varianzanalysen
Eingang	Eine Tabelle, mindestens zwei Spalten
Ausgabe	Zwei Tabellen: 1. Eingangstabelle 2. Varianzanalysetafel
Besonderheit	-

Das ANOVA–Modul (Abk. für: Analysis of Variance) berechnet eine Varianzanalysetafel mehrerer Variablen.

Es stehen zwei Betriebsarten zur Auswahl: Zum einen können mehrere Spalten einer Tabelle ausgewählt werden, die die Messreihen darstellen. Zum anderen kann eine Werte–Spalte ausgewählt werden, die durch die Werte einer Klassenspalte eingeteilt wird.

Hintergrund

Ausgangspunkt sind die Beobachtungen oder Messungen $x_{i_1}, \ldots, x_{i_{n_i}} : i \in \{1, \ldots, k\}$ k verschiedener Messreihen oder Gruppen X_i mit n_i Messwerten.

Sind die Gruppen X_i mit $N\left(\mu_i, \sigma^2\right)$ haben alle Gruppen die gleiche Varianz $\sigma^2 > 0$ und einen unbekannten Erwartungswert μ_i.

In der ANOVA werden Varianzkomponenten gebildet, die unter der Nullhypothese der Gleichheit der Erwartungswerte

$$H_0 : \mu_1 = \ldots = \mu_k$$

in den Gruppen identische Größen aufweisen müssten.

Grundlegende Größe für das weitere Vorgehen ist die Summe der Abweichungsquadrate *SAQ*

$$SAQ = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \overline{x})^2$$

vom Mittelwert

$$\overline{x} = \frac{1}{k} \sum_{i=1}^{k} \overline{x}_i$$

aller Messwerte mit dem Mittelwert der Gruppen

$$\overline{x}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \overline{x}_{ij} \ .$$

Mit der Summe der Abweichungsquadrate zwischen den Gruppen

$$SAQ_Z = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (\overline{x}_i - \overline{x})^2 = \sum_{i=1}^{k} n_i (\overline{x}_i - \overline{x})^2$$

und der Summe der Abweichungsquadrate innerhalb der Gruppen

$$SAQ_I = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \overline{x})^2$$

gilt:

$$SAQ = SAQ_I + SAQ_Z \ .$$

Mit Hilfe eines F–Tests wird das Signifikanzniveau α berechnet, zu dem die Nullhypothese H_0 identischer Mittelwerte $\mu_1, \ldots, \mu_n$ verworfen werden muss.

Mit Hilfe der Gesamtanzahl aller Beobachtungswerte

$$N = \sum_{i=1}^{k} n_i > k$$

kann der F–Test zum Vergleich der Mittelwerte μ_i verwandt werden.

Zunächst werden die so genannten mittleren Quadratsummen

$$MQ_Z = \frac{SAQ_Z}{k-1}$$

$$MQ_I = \frac{SAQ_I}{N-k}$$

aus den *SAQ*–Werten und deren Freiheitsgraden gebildet.

Wird der Wert

$$F = \frac{MQ_Z}{MQ_I}$$

größer als

$$F > F_{k-1,N-k,1-\alpha} \ ,$$

ist die Nullhypothese H_0 zum Signifikanzniveau α zu verwerfen.

Ausgabe

Die von ANOVA erzeugte Tabelle stellt die Varianztafel dar. Die Spalten *Sum of Squares*, *df* und *Mean Square* werden gemäß der oben genannten Formeln berechnet. Die Spalte *F* enthält den ermittelten F–Wert, *Sig* das Signifikanzniveau, zu dem die Hypothese H_0 gerade noch angenommen wird.

Streuungs-ursache	Quadratsum (Sum of Squares)	Freiheits-grade (df)	Mittlere Quadrat-summe (Mean Square)	F-Wert (F)	Signifikanz (Sig)
Between Groups	SAQ_Z	$k-1$	MQ_Z	F	α
Within Groups	SAQ_I	$N-k$	MQ_I		
Total	SAQ	$N-1$			

Bedienung

Um eine ANOVA durchführen zu können, wählen Sie zunächst unter Factor Selection den Arbeitsmodus des Moduls:

Columns. In dieser Einstellung werden mindestens zwei Spalten erwartet, die den Messeihen entsprechen. Es wird immer eine weiteres, leeres Eingabefeld angezeigt, in das Sie jeweils eine weitere Spalte zur Analyse hinzufügen können.

Classes. Dieser Modus erwartet zwei Spalten: die Messwerte *(Value Column)* und die Klassifizierung *(Classes Column)*. Für jeden unterschiedlichen Wert der *Values Column* wird eine neue Messreihe erzeugt, die gegen die jeweils anderen getestet wird.

Bei „Classes“ sind ohnehin Messreihen mit unterschiedlicher Anzahl von Messwerten zu erwarten, so dass NULL–Werte keinen negativen Einfluss haben.

Tipp!
Im Modus „Columns“ werden NULL–Werte eine Messreihe nicht berücksichtigt. Dadurch können jedoch Messreihen letztlich eine unterschiedliche Anzahl an Werten enthalten. Möchten Sie Werte paarweise ausschließen, dann verwenden Sie zuvor die Filterfunktion des Join Tools (siehe 4.2.4, Seite 84).

4.3.2. Base Table Statistics

Übernimmt die Quelltabelle und erzeugt zusätzlich eine Tabelle, in der die grundlegenden Eigenschaften (Min, Max, Average, Sum, Sample Variance, Population Variance, Sample Standard Deviation, Population Standard Deviation, Rows, Valid Rows) der in den Funktionsblöcken ausgewählten Spalten enthalten sind.

Kurzbeschreibung	
Verwendungszweck	Berechnung grundlegenden Eigenschaften einer Tabelle
Eingang	Eine Tabelle, eine Spalte pro Funktionsblock, nur numerische Typen
Ausgabe	Zwei Tabellen: 1. Eingangstabelle 2. Tabelle mit einer Spalte Beschreibung und je einer Spalte Ergebnisse pro Block
Besonderheit	-

Bezugnehmend zu dem Beispiel (Abbildung 4.15) ergeben sich folgende Ergebnisse für die Spalten *Fläche* und *Bevölkerungsdichte*.

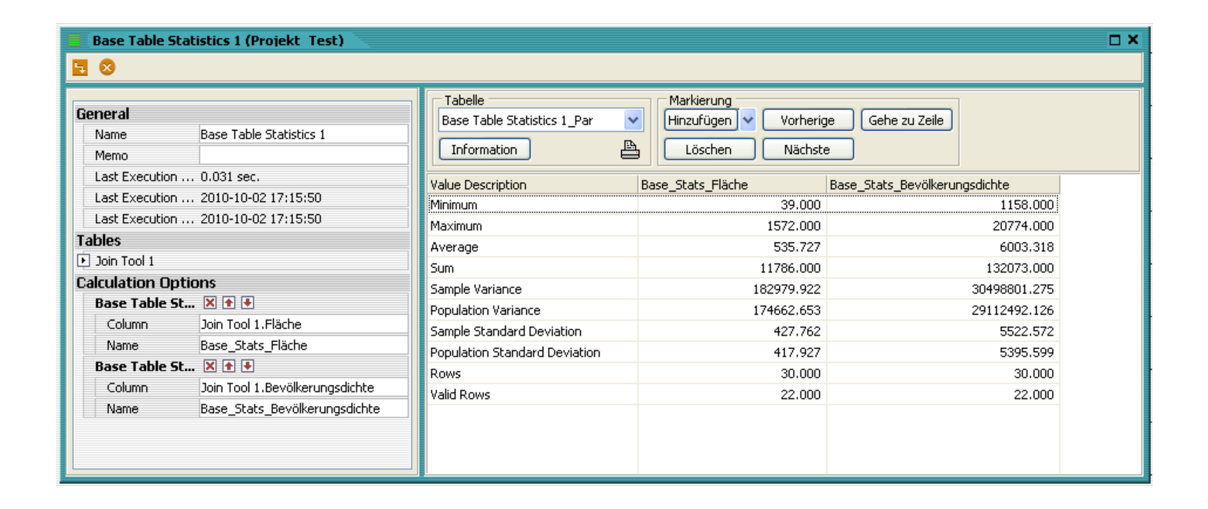


Abbildung 4.23.: Base Table Statistics

Bezeichnung	Werte
Column	Spalte, auf die sich die neu erzeugte Spalte bezieht.
Name	Name der erzeugten Spalte

Tabelle 4.3.: Parameter Base Table Statistics

4.3.3. Box Plot

Kurzbeschreibung	
Verwendungszweck	Berechnung von statistischen Kennwerten und Ausreißern
Eingang	Eine Tabelle, eine Spalte pro Funktionsblock, nur numerische Typen
Ausgabe	Zwei Tabellen: 1. Eingangstabelle 2. Tabelle mit einer Spalte Beschreibung und je einer Spalte Ergebnisse pro Block
Besonderheit	-

Berechnet die spezifischen Box–Plot–Werte der ausgewählten Spalten (Median, Hinges, Spread) und filtert gegebenenfalls die dadurch gekennzeichneten Ausreißer erster und zweiter Ordnung (Tabelle 4.4). Das Ergebnis wird mit dem Graphic–View (siehe 4.80, Seite 215) dargestellt.

Erzeugt zwei Tabellen:

1. Enthält die (ggf. gefilterten) Daten der Quelltabelle.
2. Enthält die für jede Spalte spezifischen Box–Plot–Parameter.

Bezeichnung	Werte
Boxplot Filter Options	*Filter first order outlier*: Filtert die Ausreißer erster Ordnung *Filter second order outlier*: Filtert die Ausreißer zweiter Ordnung *Filter none*: Keine Filterung

Tabelle 4.4.: Parameter Boxplot Calculation

Box–Plot–Parameter

Median. Der *Median* stellt die Mitte einer Verteilung von Werten dar. 50% aller Werte sind größer oder gleich diesem Wert; und 50% aller Werte sind kleiner oder gleich diesem Wert. Der *Median* ist bei einer ungeraden Anzahl von Werten in der Wertereihe selbst enthalten (es ist der mittlere Wert der Wertereihe), bei einer geraden Zahl von Werten berechnet er sich aus dem Durchschnitt der beiden mittleren Werte.

Hinges. Der *Lower Hinge* entspricht etwa der 25%–Percentile. Das heißt, 25% aller Werte sind kleiner oder gleich diesem Wert. Der Upper Hinge entspricht analog hierzu der 75%–Percentile.

Spread. Der *h–Spread* umfasst alle Werte zwischen Lower und Upper Hinge, also die mittleren 50% aller Werte.

Notches. Der *obere* und *untere Notch* definiert sich wie folgt:

$$Median \pm \frac{1,58 \cdot h - Spread}{\sqrt{n}}$$

wobei n die Anzahl der Werte bezeichnet.

Inner Fence. Die Grenzen des *Inner Fence* definieren sich wie folgt:

$$Lower\ Hinge - 1,5 \cdot h - Spread$$

$$Upper\ Hinge + 1,5 \cdot h - Spread$$

Outer Fence. Die Grenzen des *Outer Fence* definieren sich wie folgt:

$$Lower\ Hinge - 3 \cdot h - Spread$$

$$Upper\ Hinge + 3 \cdot h - Spread$$

Ausreißer. Die *Ausreißer erster Ordnung* („krasse Ausreißer") werden als die Werte definiert, die sich außerhalb des *Outer Fence* befinden, die *Ausreißer zweiter Ordnung* sind die Werte, die sich zwischen *Inner* und *Outer Fence* befinden.

Beispiel

Am unseren Beipiel (siehe Seite 88) berechnet das Modul für die Spalten *Fläche* und *Bevölkerungsdichte* die Boxplotwerte inlusive des *Filter first order outlier* bzw. *Filter none*.

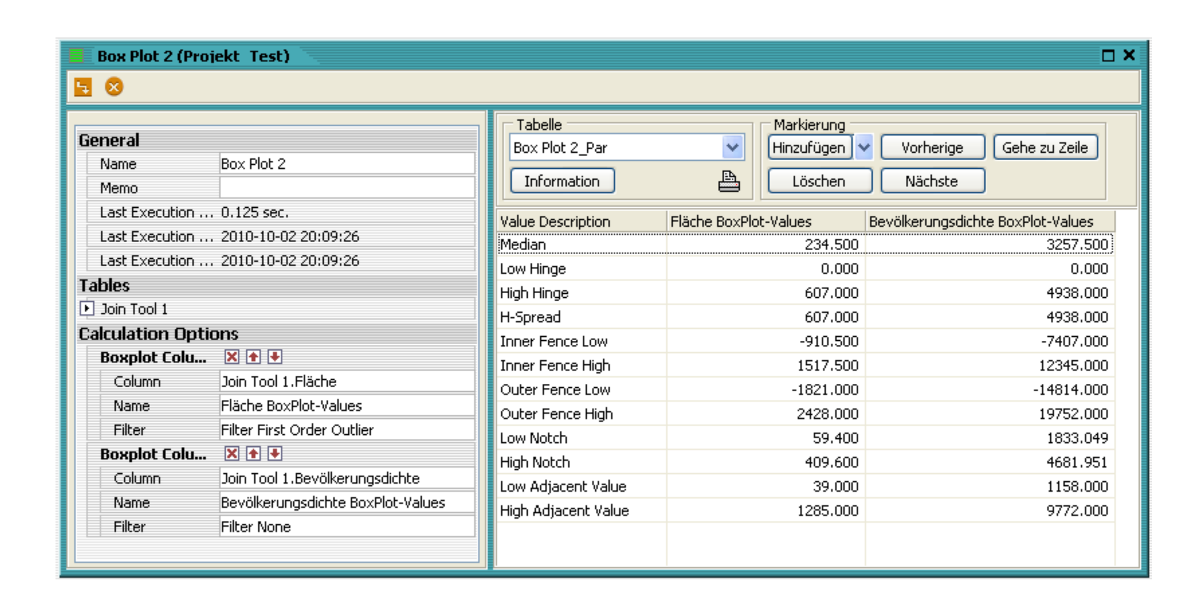


Abbildung 4.24.: Boxplotwerte des Beispiels (siehe Abbildung 4.15)

4.3.4. Chi–Square

Kurzbeschreibung	
Verwendungszweck	Bewertet die Qualität einer empirischen gegenüber einer hypothetischen Verteilung.
Eingang	Eine Tabelle, zwei Spalten als Variablen pro Funktionsblock, nur numerische Typen
Ausgabe	Zwei Tabellen: 1. Eingangstabelle zzgl. einer Ergebnisspalte pro Funktionsblock 2. Tabelle mit einer Spalte Beschreibung und je einer Spalte Ergebnisse pro ausgewählter Spalte
Besonderheit	-

Mit dem χ^2 (Chi–Square) Modul werden zwei Verteilungen mit Hilfe des χ^2–Anpassungstests miteinander verglichen.

Haupteinsatzzweck ist der Test einer empirischen Verteilung gegen eine hypothetische, um die Qualität der empirischen bewerten zu können.

Tipp!
Verwenden Sie zur Erzeugung einer hypothetischen Verteilungen das Modul Density (siehe 4.3.7, Seite 118).

Hintergrund

Sind die Häufigkeiten verschiedener Klassen einer Verteilung $h_1, \ldots, h_n$ und die Mittelwerte $m_1, \ldots, m_n$, dann ist die erwartete Häufigkeit

$$\chi^2 = \sum_{i=1}^{n} \frac{(m_i - h_i)^2}{h_i}$$

ein Maß für die Abweichung der Stichprobe von der erwarteten Häufigkeit.

Mit Hilfe des χ^2-Anpassungstests wird die Hypothese
H_0: *die Grundgesamtheit der empirischen entspricht der hypothetischen Verteilung*
gegen die Hypothese
H_1: *die Grundgesamtheit der empirischen entspricht nicht der hypothetischen Verteilung zum Signifikanzniveau α getestet.*

Für die Durchführung des Testes müssen die gegeneinander zu testenden Verteilungen in n Klassen eingeteilt und deren Häufigkeiten bekannt sein.

Die die Klassen erzeugenden Intervalle $I_1 = [i1, i2]\,, ..., I_n = [i_{n-1}, i_n]$ müssen für alle Verteilungen identisch sein. Diese Klasseneinteilung und Berechnung der den Klassen $I_1, \ldots, I_n$ entsprechenden Häufigkeiten $h_1, \ldots, h_n$ können mit dem Density–Modul (sie-

he 4.3.7, Seite 118) durchgeführt werden.

${\chi_s}^2$ ist unter der H_0 asymptotisch χ^2 verteilt mit $k-1$ Freiheitsgraden.
Die Hypothese H_0 wird zum Signifikanzniveau α verworfen, wenn

$$ {\chi_s}^2 > x_{n-1;1-\alpha} \ . $$

Die Anzahl der Freiheitsgrade n ist standardmäßig die Anzahl der Häufigkeitsintervalle minus 1.

Fur jeden aus den Daten geschatzten Parameter der hypothetischen Verteilung muss ein Freiheitsgrad abgezogen werden. Wenn demnach zum Beispiel im Density–Modul die Parameter μ und σ^2 einer Normalverteilung $N\left(\mu, \sigma^2\right)$ aus den Daten geschatzt wurden, ist fur jeden Parameter ein Freiheitsgrad abzuziehen. In diesem Fall also 2.

Anwendung

Das Chi–Square–Modul erwartet als Eingabe zwei Tabellen mit einer empirischen *(Emp. Freq.)* und einer hypothetische Verteilung *(Hyp. Freq.)*, gegen die getestet werden soll.

Weiterhin müssen die Anzahl der Freiheitsgrade *(Excess Degrees of Freedom)* festgelegt werden.

Legen Sie anschliesend noch das Signifikanz–Niveau α *(Significance)* fest, auf das die Hyptohese H_0 verworfen werden muss.

Ausgabe

Erzeugt zwei Tabellen:

1. Eingangstabelle zuzüglich einer Spalte mit der Abweichung der Stichprobe von der erwarteten Häufigkeit des jeweiligen

Intervalls nach der Formel

$$\frac{(m_i - h_i)^2}{h_i}$$

2. Parameter des Verteilungstests (Tabelle 4.5)

Spaltenname	**Bedeutung**
Emp. Freq. Col.	Name der Spalte der empirischen Häufigkeit
Hyp. Freq. Col.	Name der Spalte der hypothetischen Häufigkeit
Significance	Signifikanzniveau α
Chi–Square Value	Auf Basis der Häufigkeitsverteilung berechneter Wert ${\chi_s}^2$
Accept. Chi–Square Value	Auf Basis der Freiheitsgrade und des Signifikanzniveaus berechneter Chi–Square–Wert $\chi_{n-1;1-\alpha}$
Accept Hyp.	Gibt an („Yes“/“No“), ob auf Basis des berechneten Chi–Square–Werts zum Signifikanzniveau α die Hypothese H_0 verworfen werden muss.

Tabelle 4.5.: Ergebnistabelle Chi–Square

4.3.5. Correlation

Kurzbeschreibung	
Verwendungszweck	Berechnet die Korrelationskoeffizienten zweier Spalten.
Eingang	Eine Tabelle, mindestens zwei Spalten, nur numerische Typen
Ausgabe	Zwei Tabellen: 1. Korrelationsmatrix (zweidimensional) 2. Korrelationstabelle mit zwei Spalten Beschreibungen der Kombinationen und einer Spalte Ergebnis der jeweiligen Korrelation
Besonderheit	-

Berechnet die Korrelationskoeffizienten aller möglichen Kombinationen von jeweils zwei der ausgewählten Spalten.

Erzeugt zwei Tabellen:

1. Korrelations–Matrix (zweidimensional; kann nur mit dem Graphic–View (Seite 201) oder Matrix–View (Seite 199) angezeigt werden.)
2. Korrelations–Tabelle

Hintergrund

Der *Korrelationskoeffizient* ist ein Maß für kontinuierliche Variablen. Im Gegensatz zur Regressionsanalyse, in der es eine klare Trennung zwischen abhängiger und unabhängiger Variablen gibt, betrachtet der *Korrelationskoeffizient* beide Variablen gleichberechtigt. Es geht um Kovariation von X und Y. Darüber hinaus werden beide Variablen, die eventuell in unterschiedlichen Maßeinheiten gemessen wurden, durch eine *Z–Transformation* (siehe

4.3.21 auf Seite 167) standardisiert und vergleichbar gemacht. Der *Korrelationskoeffizient* ist eine mit den jeweiligen Standardabweichungen standardisierte *Kovarianz* zweier Variablen X und Y.

Der Wertebereich des Korrelationskoeffizienten geht von -1 bis $+1$, wobei 0 darauf hinweist, daß kein Zusammenhang zwischen beiden Variablen existiert. -1 weist auf einen perfekten negativen und $+1$ auf einen perfekten positiven Zusammenhang hin. Dabei sollte man beachten, daß der Korrelationskoeffizient nur lineare Zusammenhänge bewertet. Stehen beide Variablen in einem komplett nichtlinearen Zusammenhang, ist es gut möglich, daß der Korrelationskoeffizient gleich 0 ist.

Der Korrelationskoeffizient $Kor\left(X,Y\right)$ der Variablen X und Y lässt sich wie folgt berechnen:

$$\begin{aligned} Kor\left(X,Y\right) &= \frac{\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i-\overline{x}\right)\left(y_i-\overline{y}\right)}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i-\overline{x}\right)^2}\cdot\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(y_i-\overline{y}\right)^2}} \\ &= \frac{Cov\left(X,Y\right)}{\sqrt{Var\left(X\right)}\cdot\sqrt{Var\left(Y\right)}} \end{aligned}$$

Die Abbildungen 4.25, 4.26 und 4.27 zeigen drei Beispiele mit Korrelationen in der Nähe von 1, -1 und 0.

Je näher der Betrag des Korrelationskoeffizienten am Wert 1 liegt, desto größerer linearer Zusammenhang existiert zwischen den betrachteten Variablen.

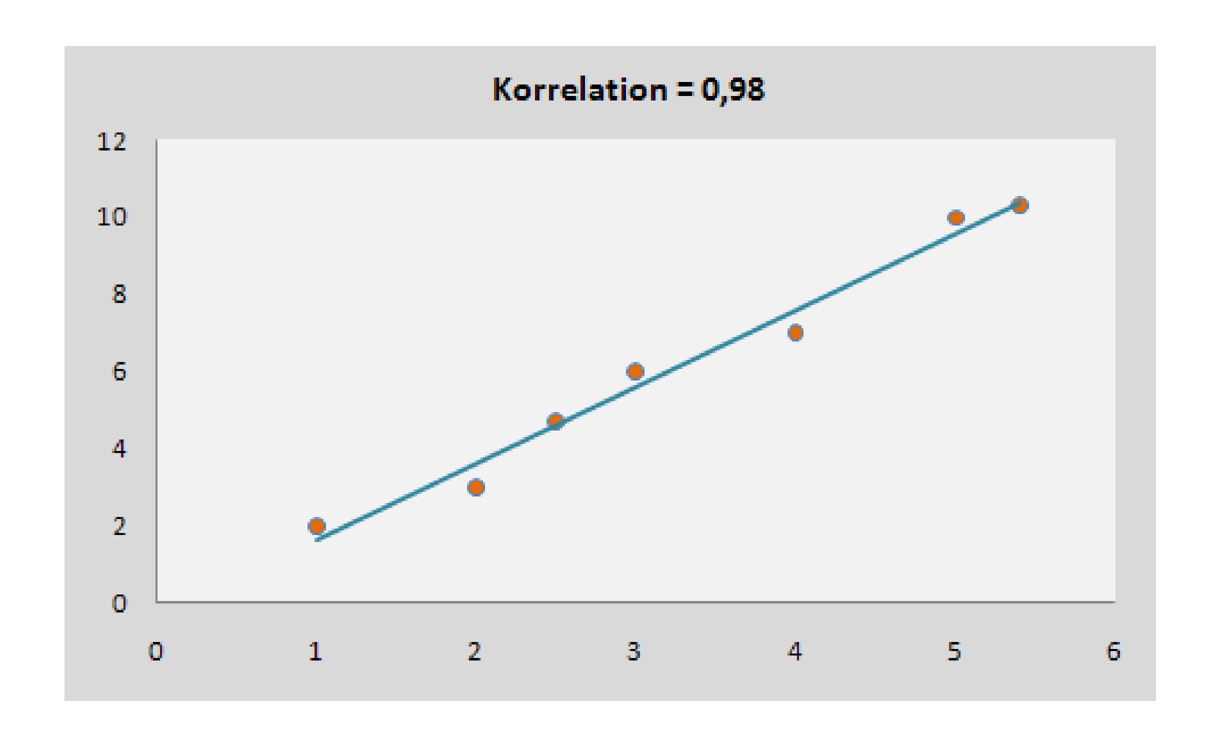


Abbildung 4.25.: Positiver fast perfekter Zusammenhang von X und Y

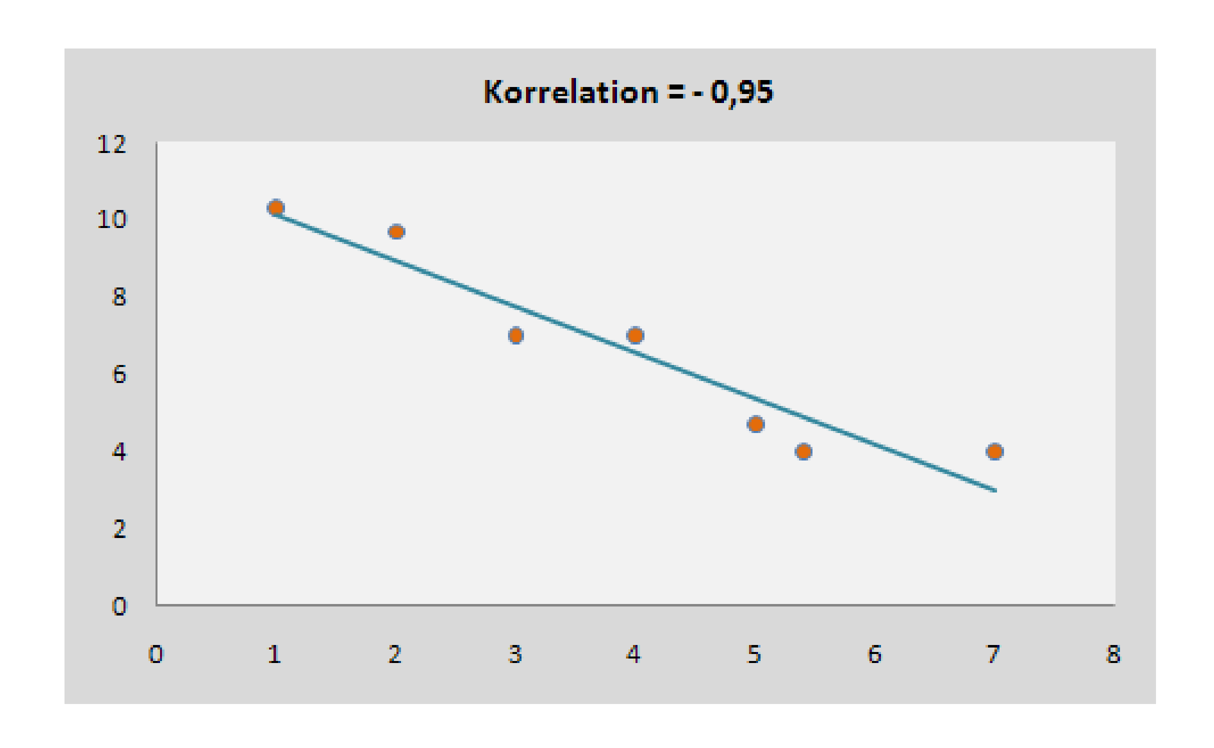


Abbildung 4.26.: Negativer fast perfekter Zusammenhang von X und Y

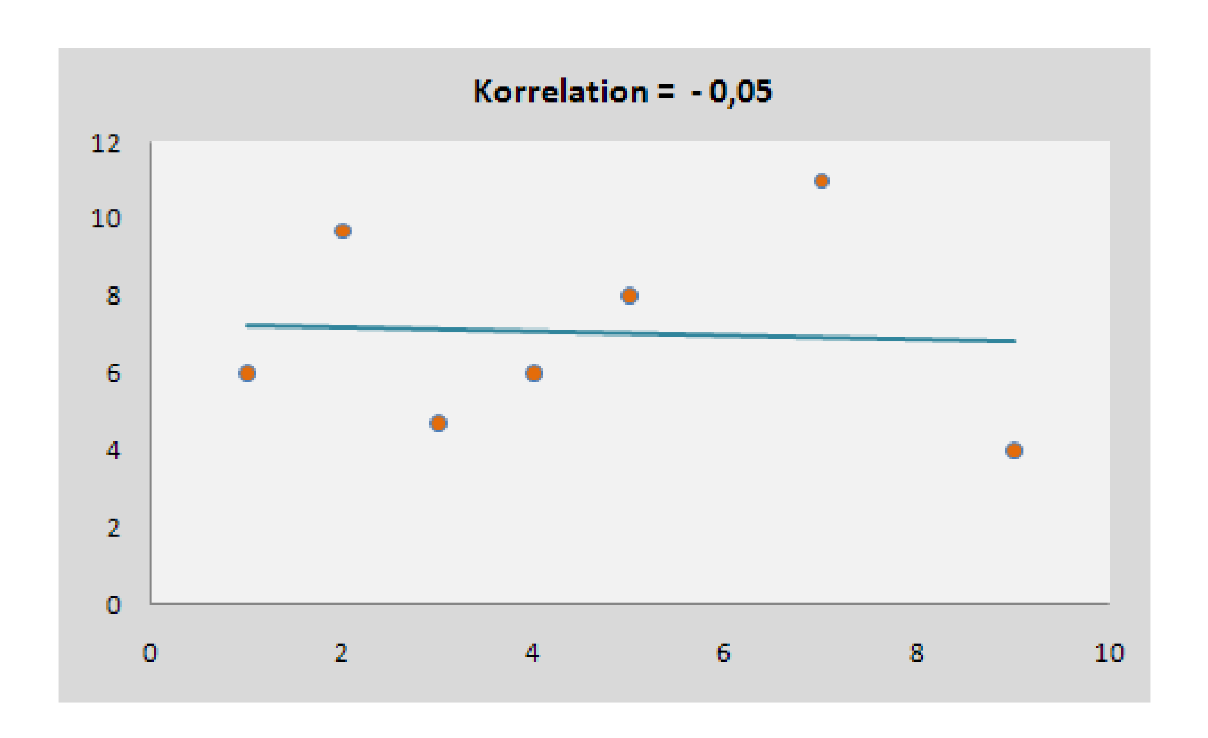


Abbildung 4.27.: Nichtlinearer Zusammenhang von X und Y

Beispiel

Wie verhält sich die Einwohnerzahl zur Fläche?
Zu erwarten ist eine positive Korrelation, da je größer die Einwohnerzahl in der Stadt ist, desto mehr Fläche wird auch benötigt.

Die berechnete Korrelation (siehe Abbildung 4.28) beträgt 0.717.

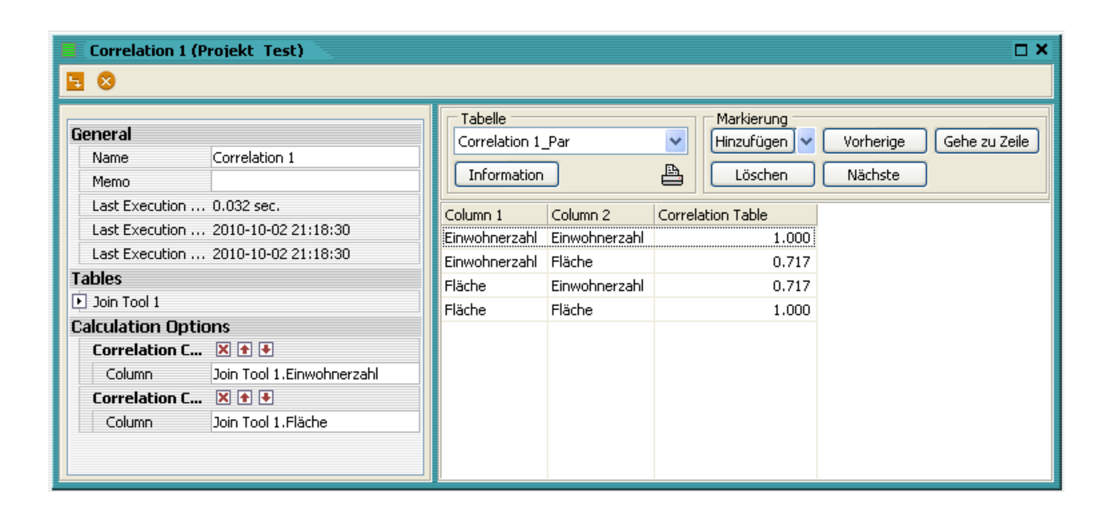


Abbildung 4.28.: Zusammenhang von *Einwohnerzahl* und *Fläche*

4.3.6. Custom

Kurzbeschreibung	
Verwendungszweck	Berechnung frei definierbarer Formeln
Eingang	Eine Tabelle, beiliebig viele Spalten beliebigen Typs, Verwendung abhängig von der definierten Berechnungsformel
Ausgabe	Eingangstabelle zzgl. je einer Spalte pro Funktionsblock
Besonderheit	Die Ausgabe muss nicht zwangsläufig mit den Eingangsdaten in Verbindung stehen, da mithilfe von Funktionen Bezüge auf andere Informationen hergestellt werden können.

Generiert gemäß einer frei definierbaren Formel neue Spalten aus den Quelldaten.

Das Modul erzeugt eine Tabelle, die die Spalten der eingehenden Tabelle zuzüglich je einer Ergebnisspalte pro Funktionsblock enthält.

Formel–Editor

Der Editor (Abbildung 4.29) unterstützt Sie bei der Eingabe komplexer mathematischer Formeln. Dabei überprüft das System im Hintergrund die Gültigkeit des Terms und weist Sie auf eventuelle Fehler hin.

In der linken Tabelle *Columns* (1) erhalten Sie eine Übersicht aller zur Verfügung stehenden Spaltennamen. Die rechte Tabelle *Functions* (2) zeigt Ihnen die auf dem System zur Verfügung stehenden Berechnungsfunktionen nebst einer kurzen Beschreibung ihrer Funktionsweise und Syntax.

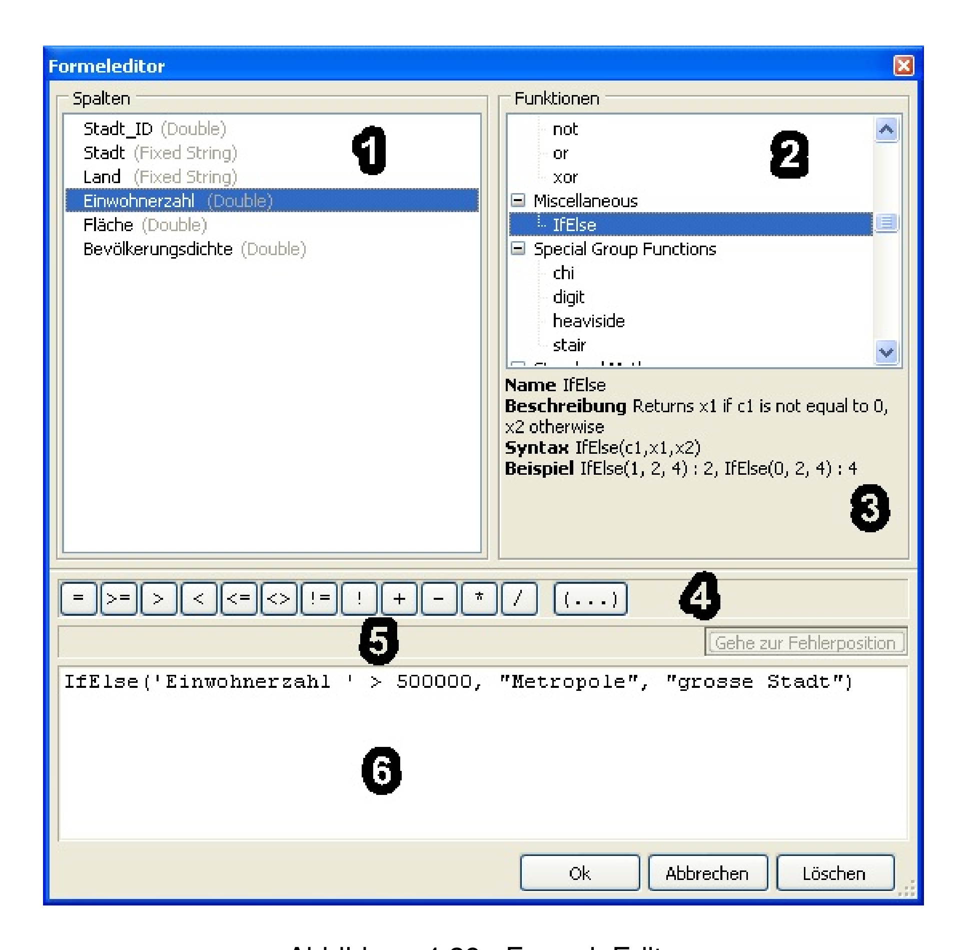


Abbildung 4.29.: Formel–Editor

1. Verfügbare Spalten
2. Funktionen
3. Kurzbeschreibung und Beispiele der Funktionen
4. Operatoren
5. Ergebnis der Syntax–Überprüfung
6. Formel–Editor

Tipp!
Aus beiden Tabellen lassen sich die Elemente direkt mit der Maus in das Eingabefeld ziehen oder per Doppelklick an die Eingabemarke setzen.

Ihnen stehen alle in der Tool–Leiste (4) aufgeführten mathematischen Operatoren zur Verfügung. Sie können sie entweder durch Drücken des jeweiligen Buttons in das Eingabefeld einfügen oder

direkt über die Tastatur eingeben. Zum einfachen Klammern eines mehrere Anweisungen umfassenden Ausdrucks markieren Sie den entsprechenden Abschnitt und wählen die Schaltfläche „( )“.

Tipp!
Bei Spaltennamen müssen Sie unbedingt die Groß-/Kleinschreibung beachten. Enthalten Namen Leerzeichen oder Punkte, müssen sie zudem in einfache Anführungszeichen gesetzt werden (z.B. „My.Special Column“).

Wird ein Fehler in der Formel erkannt, so erfolgt eine Beschreibung im Anzeigebereich oberhalb des Eingabefensters. Mit der Funktion *Go to errorposition* markieren Sie die vom System nicht interpretierbare Zeichenfolge.

Funktions–Parser

Die Rechnenfunktionalität des Custom–Moduls beruht auf einen Funktions–Parser, der die eingegebenen Formeln interpretiert und berechnet.

Der Funktions–Parser führt die eingegebene Berechnung für jede Zeile der ausgewählten Tabelle aus. Das Ergebnis wird in einer neuen Spalte abgelegt. Die Formel kann sich dabei auf beliebige Spalten der Tabelle beziehen oder eine davon unabhängige Berechnung ausführen.

Eine vollständige Liste aller vom Funktions–Parser zur Verfügung gestellten Operatoren und Funktionen finden Sie im Anhang C ab Seite 245.

Formelsyntax

Die Formeln im Custom–Modul müssen eine spezielle Schreibweise („Syntax") aufweisen. Operanden eines Operators oder Parameter einer Funktion können Konstanten oder Spalten derselben Zeile einer Tabelle sein. Die Berechnung wird für jede Zeile einzeln ausgeführt, bezieht sich demnach auch immer auf die Spaltenwerte der aktuellen Zeile.

Die Funktionen haben folgende Syntax:
$Funktionsname\,(Parameter1, Parameter2, \ldots)$

Die Parameter einer Funktion müssen durch Kommata getrennt sein und können wiederum Funktionen oder Operationen enthalten.

Beispiele

- $IfElse\,(x = y, 42, 48)$
  Operationen haben folgende Syntax:
  $Operator\ Operand$
- $(x < 5)\ or\ (y > 4)$
  $Operand\ Operator\ Operand$
- *"Hallo Welt!"*
  Zeichenfolgen („Strings") werden grundsätzlich in doppelte Hochkommata gesetzt.
- *'Tabelle.Spalte'*
  Spalten– und Tabellennamen müssen, sofern sie Sonderzeichen wie z.B. ein Leerzeichen, „%" oder „&" enthalten, in einfache Hochkommata gesetzt werden. Tabellenname und Spaltenname werden durch einen Punkt getrennt.

Berechnungsreihenfolge

RayQ führt die Rechenoperationen in Übereinstimmung mit der Reihenfolge der Funktionen von links nach rechts aus. Die Reihenfolge der Rechenschritte kann mit Hilfe von Klammern festge-

legt werden.

Stehen verschiedene Operatoren in einer Formel, werden diese in der folgenden Reihenfolge berechnet, sofern keine Klammern gesetzt sind:

1. Multiplikation bzw. Division (*, /)
2. Addition bzw. Subtraktion (+, -)
3. Vergleichsoperatoren (=, <, <=, >, >=, <>)
4. Logische Operatoren (OR, AND, XOR, NOT)

Spaltentypen

Der Funktions-Parser erzeugt abhängig von der verwendeten Formel und Spalte standardmäßig die folgenden Spaltentypen: Double, Integer, Bool, String und Datetime.

Möchten Sie einen bestimmten Typen festlegen, stehen Ihnen spezielle Wandlungsfunktionen zur Verfügung. Grundsätzlich kann jeder Typ in jeden umgewandelt werden. Die bei der Konvertierung verwendeten Regeln entnehmen Sie der Tabelle 4.6.

> **Wichtig!**
> *Dezimalstellen*. Zur Abtrennung von Dezimalstellen verwendet RayQ einheitlich den international gebräuchlichen Punkt, also z.B. „1.5“ statt „1,5“.

von \ nach	Double	Integer	Bool	String	Datetime
Double	$=$	Nachkommastellen werden abgeschnitten	Nachkommastellen werden abgeschnitten, $0 = falsch$, sonst $= wahr$.	Zahl als Text	Nachkommastellen werden abgeschnitten, $0 = 15.10.1582$, $+1 = +1s$
Integer	$=$	$=$	$0 = falsch$, sonst $= wahr$	Zahl als Text	$0 = 15.10.1582$, $+1 = +1s$
Bool	$falsch = 0.0$, $wahr = 1.0$	$falsch = 0$, $wahr = 1$	$=$	$falsch =$ „0“, $wahr =$ „1“	$falsch = 15.10.1582$, $wahr = 15.10.1582, 1s$
String	wenn keine reine Zahl als Text, dann $= 0.0$	wenn keine reine Zahl als Text, dann $= 0$	„0“ oder keine Zahl im Text, dann $= falsch$, sonst $= wahr$	$=$	wenn kein Datum im Text erkannt, dann $= 15.10.1582$
Date-Time	$15.10.1582 = 0$, $+1s = +1.0$	$15.10.1582 = 0$, $+1s = +1$	15.10.1582 $= falsch$, sonst $= wahr$	Datum als Text im Format z.B. $01.03.2003$ $10:42:00$	$=$

Tabelle 4.6.: Typkonvertierungen

Gemischte Typen. Die meisten Funktionen setzen einen bestimmten Datentyp voraus, z.B. Double oder Integer bei Berechnungen. Verwenden Sie hier andere Typen, werden diese automatisch in den erwarteten umgewandelt.

Werden in einer Formel verschiedene Typen miteinander vermischt, werden die Typen in den jeweils genaueren Typ umgewandelt, also z.B. Integer in Double.

Datum. Der Datumstyp bezieht sich auf den Anfang des Gregorianischen Kalenders, dem 15. Oktober 1582. Als Zahl ausgedrückt entspricht dieses Datum dem Wert „0“.

Bei der Konvertierung eines in Textform geschriebenen Datums in den Datumstypen wird jede gebräuchliche Schreibweise erkannt. Eine Liste der unterstützten Formate finden sie unter C.3.4 Datum auf Seite 261.

NULL. Ein NULL–Wert ist ein nicht definierter Wert, z.B. wenn eine Zeile einer Spalte „leer“ ist. Wird ein NULL–Wert innerhalb einer Funktion verwendet, kann diese nicht ausgeführt werden, weshalb wiederum NULL zurückgegeben wird. Einzige Ausnahme ist hier die Funktion *IsNull()*, die zur Überprüfung auf NULL–Werte vorgesehen ist (siehe C.3.1, Seite 251).

4.3.7. Density

Kurzbeschreibung	
Verwendungszweck	Berechnung der Dichteverteilung
Eingang	Eine Tabelle, eine Spalte als Variable pro Funktionsblock, nur numerische Typen
Ausgabe	Zwei Tabellen: 1. Eingangstabelle zzgl. einer Ergebnisspalte 2. Ergebnistabelle, vier bzw. fünf Spalten mit Beschreibung der Intervalle einer Spalte, der empirischer Verteilung und optional einer theoretischen Verteilung
Besonderheit	-

Berechnet die empirische absolute Häufigkeitsverteilung einer Spalte und optional die entsprechende einer theoretischen Verteilung. Die Häufigkeitsverteilungen werden auf Basis von Intervallen be-

rechnet.

Die Intervallgrenzen werden entweder aus der in *Number of Intervalls* angegebenen Anzahl von Intervallen aus den Werten in der ausgewählten Spalte berechnet oder durch manuelle Angabe der Intervallgrenzen nach Anwahl von *Custom Intervalls* festgelegt.

Werden die Intervalle automatisch berechnet, sind alle Intervallbreiten identisch. Mit *From Data* kann festgelegt werden, ob die untere Intervallgrenze des ersten Intervalls und die obere Grenze des letzten Intervalls aus den zugrunde liegenden Daten bestimmt oder aber manuell über *Minimum* bzw. *Maximum* festgelegt werden sollen.

Bezeichnung	**Werte**
Custom Intervals	Automatische oder manuelle Festlegung der Intervallgrenzen
Number of Intervals (automatisch)	Anzahl der Intervalle
From Data (automatisch)	Bestimmung der unteren Intervallgrenze des ersten und der oberen Intervallgrenze des letzten Intervalls aus den Daten
Minimum (automatisch)	Untere Intervallgrenze des ersten Intervalls
Maximum (automatisch)	Obere Intervallgrenze des letzten Intervalls
I (manuell)	Einzelne Intervallgrenzen

Tabelle 4.7.: Density: Interval Definition

Sollen die Minimum und Maximum manuell festgelegt werden, kann der Wert in diesem Modus editert werden.

Standardmäßig wird keine theoretische absolute Häufigkeitsverteilung berechnet. Sollen diese — z.B. als Vorbereitung für einen Chi–Square–Anpassungstest — auch berechnet werden, wählen

Sie unter Distribution die gewünschte Verteilung aus.

Ist *Estimate* aktiviert, werden die Verteilungsparameter aus der ausgewählten Spalte berechnet. Ansonsten können die Parameter auch manuell festgelegt werden.

Unter *Calculation Options* wählen Sie die Spalten aus, von der die theoretischen bzw. empirischen Häufigkeitsverteilungen berechnet werden sollen.

Erzeugt zwei Tabellen:

1. Entspricht der Quelltabelle

2. Tabelle mit den Häufigkeitsverteilungen

Bezeichnung	**Werte**
Distribution	Verteilungsfunktion (Normal, Expronential, Poisson, Uniform)
Estimate (automatisch)	Parameter werden aus den zugrunde liegenden Daten berechnet
Mean („Normal“)	Mittelwert
Standard Deviation („Normal“)	Standardabweichung
Decay Factor („Exponential“)	Ausfallrate
Minimum („Exponential“)	Verschiebt den Nullpunkt der Verteilung
Lambda („Poisson“)	Erwartungswert der Poissonverteilung
Lower Bound („Uniform“)	Untere Grenze
Upper Bound („Uniform“)	Obere Grenze

Tabelle 4.8.: Density: Calculation Options

Bezeichnung	**Werte**
Column	Gibt die Spalte an, auf die sich die neu zu erzeugende Spalte beziehen soll
Name of Interval Column	Name der erzeugten Spalte, die die Intervall–Definitionen enthält.
Name of lower limit column	Name der erzeugten Spalte, die die Untergrenzen der Intervalle enthält.
Name of upper limit column	Name der erzeugten Spalte, die die Obergrenzen der Intervalle enthält.
Name of Count Column	Name der erzeugten Spalte, die die Häufigkeiten enthält.
Range	Legt fest, wie der Wertebereich gewählt werden soll, in dem die Häufigkeiten berechnet werden. Bei der Option *Auto* wird als untere Grenze das Minimum, als obere Grenze das Maximum der Quelldaten gewählt. Bei der Option *Custom* können unter Min bzw. Max diese Werte selbst eingegeben werden.

Tabelle 4.9.: Parameter Density

Beipiel

In unserem Beispiel soll überprüft werden, ob die Einwohnerzahl-Verteilung einer Exponentialverteilung entspricht.
Die Spalte *Einwohnerzahl_Density* gibt die tatsächliche Anzahl der Werte in den einzelnen Intervallen an, während die Spalte *Distribution_Frequencies* die theoretische Anzahl einer Exponentialverteilung angibt.

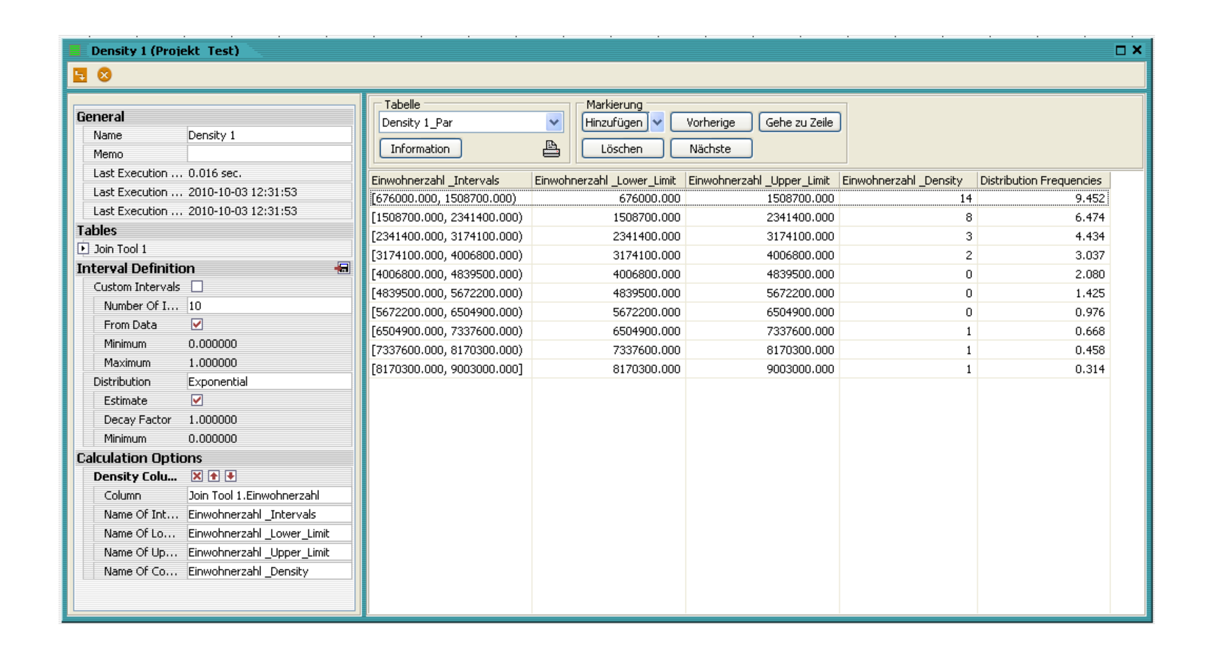


Abbildung 4.30.: Beispiel: Density–Modul

4.3.8. Disaggregation

Kurzbeschreibung	
Verwendungszweck	Disaggregation von aggregierten Zeichenketten
Eingang	Eine Tabelle, die eine aggregierte Spalte enthält
Ausgabe	Zwei Tabellen: 1. Eingangstabelle 2. Ergebnistabelle, zzgl. einer Ergebnisspalte
Besonderheit	-

Enthält eine Tabelle String–aggregierte Spalte (Siehe Group – Modul, Seite 126), so lassen sich die einzelnen Elemente dieser Spalte mit Das Modul Disaggregation in die Spaltenfolrm bringen.

Jede String–aggregierte Zelle mit k Elementen erscheint in der

Ergebnistabelle k–mal. In einer zusätzlichen Spalte werden die einzelnen Elemente aufgezählt.

Funktion	**Bezeichnung**
Use All Columns	Auswahl aller Spalten
Use Disaggregations As One Column	Auswahl der Aggregation in einer Spalte
Delete Rows With Empty Fields	Löschung der Reihen mit leeren Einträgen
Enable Group Marking	Auswahl der Markierungsfunktion

Tabelle 4.10.: Disaggregation: Disaggregation Options

Funktion	**Bezeichnung**
Data Column	
Column	Auswahl der Spalte, in der die Daten aggregiert sind.
Name	Name der Spalte, in der die Daten aggregiert sind.
Aggregation Column	
Column	Auswahl der Spalte, die disaggregiert werden soll.
Name	Name der Spalte, die disaggregiert werden soll.
Delimeter	Trennzeichen in der aggregierten Spalte
Length	Länge der Zeichenkette der neuen disaggregierten Spalte der Spalte, die disaggregiert werden soll.

Tabelle 4.11.: Disaggregation: Calculation Options

Beispiel

Die Tabelle in der Abbildung 4.31 soll disaggregiert werden.

Spalte
A01;A02;A03
A01;A03
A02;A03
A03;A04
A01
A03

Abbildung 4.31.: Beispiel Aggregation

Die Einstellungen des Disaggregation–Moduls sind in der Abbildung 4.32 abgebildet.

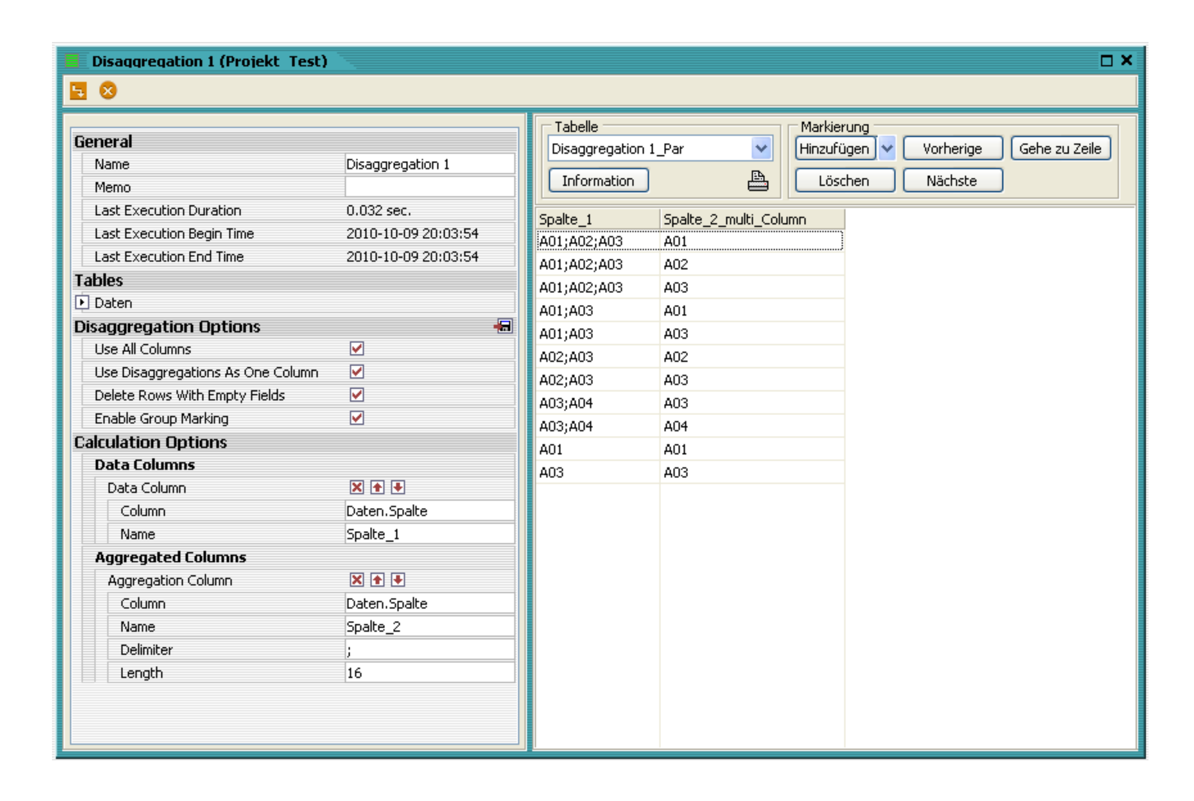


Abbildung 4.32.: Einstellungen und Ergebnis

Zu jedem Eintrag in der Ausgangstabelle (Abbildung 4.31) wird ein Block erstellt (Abbildung 4.33), der genau so viele Datensätze enthält wie der Anzahl der Elemente im Eintrag.

Spalte_1	Spalte_2_multi_Column
A01;A02;A03	A01
A01;A02;A03	A02
A01;A02;A03	A03
A01;A03	A01
A01;A03	A03
A02;A03	A02
A02;A03	A03
A03;A04	A03
A03;A04	A04
A01	A01
A03	A03

Abbildung 4.33.: Ergebnisinterpretation

4.3.9. Group

Kurzbeschreibung	
Verwendungszweck	Fasst mehrfach vorkommende Werte einer Spalte zu Gruppen zusammen und berechnet Aggregate.
Eingang	Eine Tabelle, eine Spalte beliebigen Typs pro Gruppierungsebene, eine Spalte numerischen Typs als Variable pro Aggregation
Ausgabe	Zwei Tabellen: 1. Eingangstabelle 2. Tabelle mit je einer Spalte pro Gruppierungsebene und eine pro Aggregation.
Besonderheit	Markierungen innerhalb der gruppierten Tabelle lassen sich wahlweise auf die Eingangstabelle übertragen.

Mithilfe dieses Moduls werden mehrfach vorkommende Werte einer Spalte zu Gruppen zusammengefasst. Als Ergebnis wird eine

Tabelle generiert, die Spalten mit den gruppierten Werten und in weiteren Spalte die aggregierten Werte der jeweiligen Gruppen enthält.

Mit der Option *Enable Group Marking* können Sie festlegen, ob Markierungen innerhalb der ausgegebenen, aggregierten Daten in die Quell–Tabelle übernommen werden sollen.

Mit den Funktionsblöcken *Group By* legen Sie fest, nach welchen Spalten gruppiert werden soll. Mehrere *Group–By*–Blöcke nacheinander geben entsprechend ihrer Reihenfolge eine Untergruppierung an.

Bezeichnung	**Werte**
Column	Spalte, nach der die Gruppierung durchgeführt werden soll
Name	Name der zu erzeugenden Spalte

Tabelle 4.12.: Parameter Group By

Bezeichnung	**Werte**
Kind	Bestimmt die Art der Aggregation (Count, Count Valid, Sum, Average, Min, Max, Variance, Standard Deviation, Quad Sum, Quad Average, String Aggregation).
Column	Spalte, in der die Aggregation stattfinden soll
Name	Name der zu erzeugenden Spalte
Group Level	Legt die Gruppierungsebene fest, auf die sich die Aggregation beziehen soll.

Tabelle 4.13.: Parameter Aggregation

Mit den Funktionsblöcken *Aggregations* lassen sich zusätzlich Berechnungen innerhalb der Gruppen durchführen. Sie können parallel beliebig viele Aggregationen auf Spalten durchführen. Eine

Ausnahme hierzu bildet die Aggregation *String Aggregation* — sie kann nicht in Kombination mit anderen gewählt werden — sondern nur einzeln.

Werden die Daten anhand mehrerer Spalten gruppiert, dann kann in der Aggregation festgelegt werden, auf welcher Gruppierungsebene die Aggregation berechnet werden soll. Das ermöglicht trotz Untergruppen die Aggregation auch auf einer höheren Gruppenebene durchzuführen. Die aggregierten Werte sind dann in allen Untergruppen einer Gruppierungsebene identisch.

Beispiel

Betrachten wir die folgende Tabelle einer Verkaufsabteilung:

Produkt	Zustand	Preis
A	neu	100
B	gebraucht	36
C	neu	590
C	gebraucht	253
A	gebraucht	68
B	neu	45
C	gebraucht	343
A	neu	120
C	neu	650
B	neu	42
B	gebraucht	21
A	neu	96
A	neu	88
C	gebraucht	612
B	neu	61
A	gebraucht	55

Abbildung 4.34.: Beispiel für das Group–Modul

Es sollen nun folgende Kennzahlen berechnet werden:

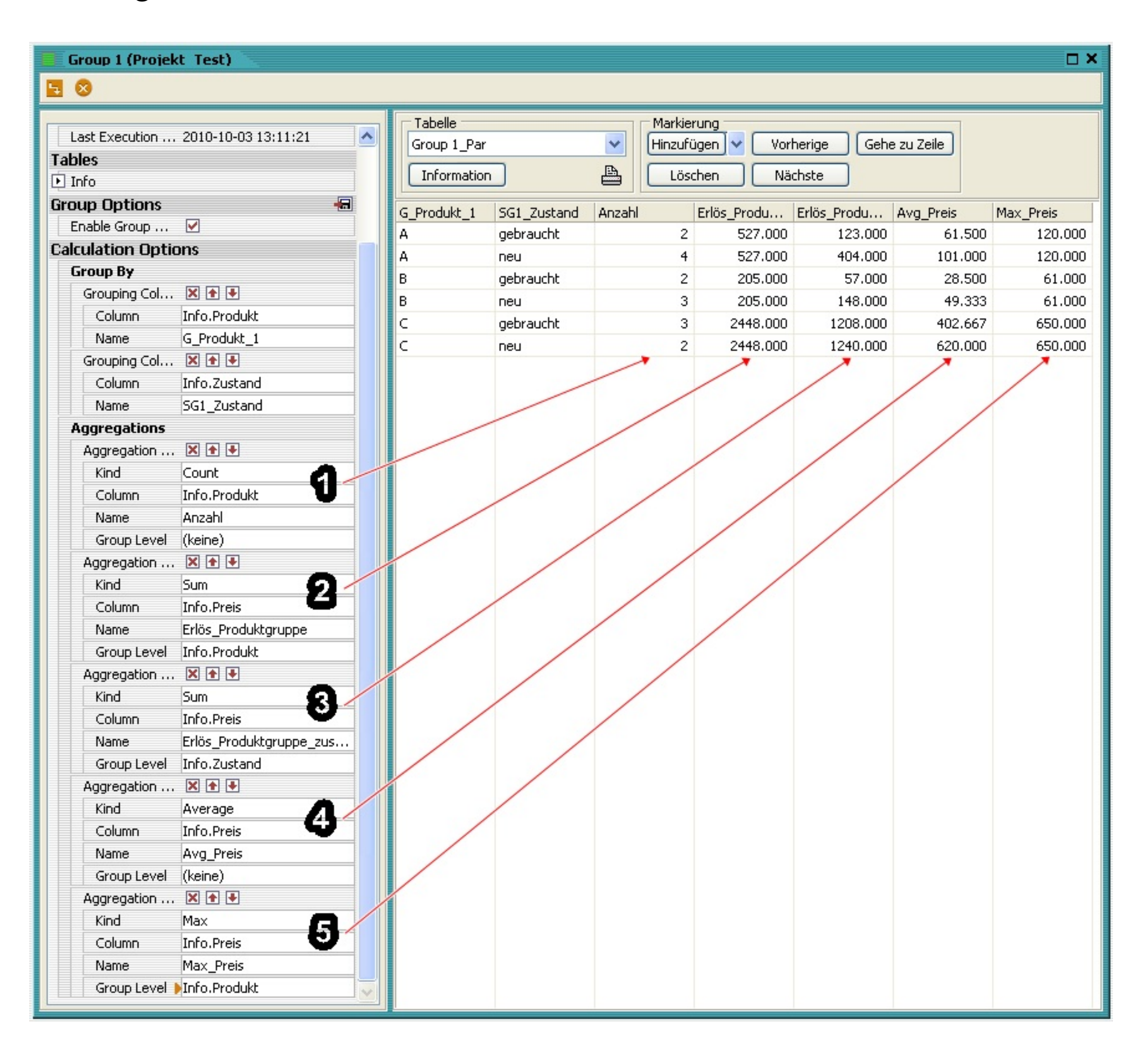
1. Anzahl verkaufter Produkte je Produktgruppe und Zustand
2. Erlös je Produktgruppe

3. Erlös je Produktgruppe und Zustand

4. Durchschnittspreis je Produktgruppe und Zustand

5. Teuersten Verkauf je Produktgruppe

Als Grouppierungsebenen dienen in diesem Beispiel die Spalten *Produktgruppe* und *Zustand*. Bei einer Aggregation auf der Ebene *Produktgruppe* muss der *Group Level* auf *Produktgruppe* eingestellt werden — analog bei einer Aggregation auf der Ebene *Produktgruppe* und *Zustand* wird der *Group Level* auf *Zustand* gesetzt.
Lösungen:



G_Produkt_1	SG1_Zustand	Anzahl	Erlös_Produ...	Erlös_Produ...	Avg_Preis	Max_Preis
A	gebraucht	2	527.000	123.000	61.500	120.000
A	neu	4	527.000	404.000	101.000	120.000
B	gebraucht	2	205.000	57.000	28.500	61.000
B	neu	3	205.000	148.000	49.333	61.000
C	gebraucht	3	2448.000	1208.000	402.667	650.000
C	neu	2	2448.000	1240.000	620.000	650.000

Abbildung 4.35.: Lösungen 1-5

4.3.10. Group–Mark

Kurzbeschreibung	
Verwendungszweck	Zeilenmarkierungenals Werte in die Tabelle übernehmen
Eingang	Eine Tabelle
Ausgabe	Eingangstabelle zzgl. je einer Spalte mit den Markierungsinformationen
Besonderheit	-

Mithilfe dieses Moduls können aus Markierungen beliebige vom Benutzer vorgegebene Werte in einer neue Spalte erzeugt und damit eine manuelle Klassifizierung der Daten vorgenommen werden. Weiterhin können NULL–Werte dieser neu erzeugten Spalte mit anderen Werten belegt werden.

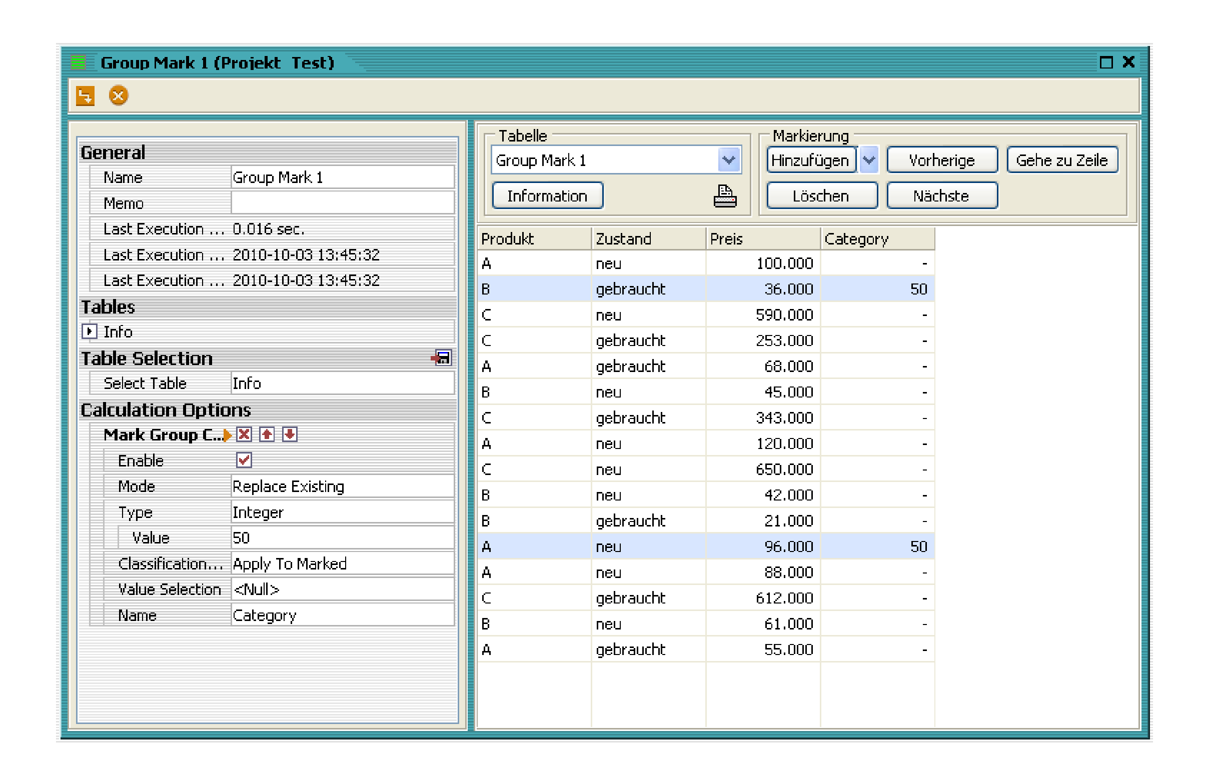


Abbildung 4.36.: Beispiel Group Mark Modul

Geben Sie unter *Select Table* die Tabelle an, die als Quelle verwendet werden soll.

Abbildung 4.36 zeigt eine mögliche Anwendung des Group Mark Moduls.

Bezeichnung	Werte
Enabled	Schaltet die Verarbeitung dieses Funktionsblocks ein und aus.
Mode	Legt fest, wie die Werte aus der aktuellen Klassifizierung erzeugt werden: *Replace Existing* setzt den eingestellten Wert überall in der aktuellen Klassifizierung durch Überschreiben der vorhandenen Werte in der neuen Spalte. *Keep Existing* setzt den eingestellten Wert in der aktuellen Klassifizierung nur in den Feldern, die nicht NULL und damit noch nicht mit dem Klassenwert belegt sind. *Remove* setzt alle Felder der aktuellen Klassifizierung auf NULL. *Add* addiert den eingestellten Wert in der aktuellen Klassifizierung zu bereits bestehenden Werten. Bei numerischen Spaltentypen ist dies die normale Addition, bei Stringtypen werden die Strings verkettet.
Type	Legt den Typen der erzeugten Spalte fest
Value	Bestimmt den Wert, mit dem die aktuelle Klassifizierung durchgeführt wird.

Classification Mode	Bestimmt die Klasse von Zeilen die der aktuelle Klassifizierung zu Grunde liegen: *Apply To Marked*: alle markierten Zeilen. *Apply To Non Marked*: alle nicht markierten Zeilen. *Apply To All Null–Values*: Zeilen, in denen der Wert NULL ist.
Value Selection	Bietet als Arbeitshilfe bei bereits erfolgten Klassifizierungen eine Liste von bereits benutzten Werten, die bei Auswahl dann in das Value–Feld übernommen werden.
Name	Name der zu erzeugenden Spalte

Tabelle 4.14.: Parameter Group Mark

4.3.11. K–Means

Kurzbezeichnung	
Verwendungszweck	Iterativer Clusteralgorithmus zum Auffinden ähnlicher Datensätze
Eingang	Beliebige Tabelle
Ausgabe	Eingangstabelle zzgl. Spalte mit Indizes der ermittelten Klassen
Besonderheit	-

Mithilfe des iterativen Clusteralgorithmus K–Means lassen sich Gruppen ähnlicher Datensätze innerhalb einer Tabelle ermitteln.

Für jede Spalte, die Sie in den Klassifikationsprozess einbeziehen möchten, legen Sie einen neuen Funktionsblock *Classification Column* an. Mit dem Parameter *Iterations* geben Sie die Anzahl der Iterationsschritte des Algorithmus an, mit *Classes* die Anzahl der zu bildenen Gruppen.

Das Modul erzeugt eine neue Tabelle mit allen Eingangsspalten zuzüglich einer neuen Spalte, die für jede Zeile die Nummer der zugeordneten Gruppe angibt.

Funktionsweise

K–Means ist ein Clusteralgorithmus, der N Datensätze in K disjunkte Cluster K_j einteilt, indem die Summe der Abstandquadrate S der Datensätze von den Clusterzentren k_j iterativ minimiert wird:

$$S = \sum_{j=1}^{K} \sum_{n \in K_j} \|x_n - k_j\|^2$$

$x_n \in X$ ist der Vektor, der den n–ten Datensatz der zu clusternden

Daten X repräsentiert, k_j ist das geometrische Zentrum

$$k_j = \frac{1}{\sharp K_j} \sum_{n \in K_j} x_n$$

des Clusters K_j und der Anzahl $\sharp K_j$ der x_n in K_j.

Der Algorithmus arbeitet mit der festen Clusteranzahl K und I Iterationsschritten, die vom Benutzer festgelegt werden müssen (Parameter *Classes* und *Iterations*).

Schritt 1. K–Means bildet zu Beginn fur jede Gruppe ein Zentrum im geometrischen Raum, der so viele Dimensionen wie Variablen (hier: Spalten) hat, die zur Ahnlichkeitsberechnung der Datensatze herangezogen werden. Das heist, K Clusterzentren k_j werden aus den vorhandenen x_n durch zufallige Auswahl aus X bestimmt $(k_j \in X)$.

Schritt 2. Alle Datensatze werden jeweils derjenigen Gruppe zugeordnet, deren Zentrum ihnen am nachsten liegt. Damit werden die Datensatze x_n dem Cluster K_j zugeordnet, zu dem der euklidische Abstand $\|x_n - k_j\|$ minimal ist.

Schritt 3. Sind alle Datensatze ihren Zentren zugeordnet, werden alle Abstande der Datensatze von ihren Gruppenzentren zu einem Kennwert aufaddiert, der die Qualitat der Gruppierungslosung ausdruckt. Diesen Kennwert versucht K–Means im Folgenden zu optimieren, indem es in der festgelegten Anzahl von Iterationsschritten die Gruppenzentren so verschiebt, dass die einzelnen Datensatze sich in jedem Schritt neuen Gruppenzentren anschliesen konnen und so ihren Abstand zu ihrem Gruppenzentrum verringern konnen.

Ein neues Clusterzentrum

$$k_j^{i+1} = \frac{1}{\#K_j^i} \sum_{n \in K_j^i} x_n$$

wird bestimmt. K_j^i sind die Cluster im i–ten Iterationsschritt und K_j^{i+1} die neuen Zentren im $(i+1)$–ten Iterationsschritt.

Die Iteration wird abgebrochen, wenn die Anzahl der Iterationsschritte erreicht ist oder sich die Zuordnung der x_n zu den k_j nicht mehr ändert.

In der vom Benutzer festgelegten Anzahl von Iterationsschritten I wird S minimiert, indem mit den neuen Clustermitten bei Schritt 2 wieder begonnen wird.

Hinweis!
Eine prinzipielle Schwäche des K–Means–Algorithmus besteht darin, dass das erreichte Minimum von S gegebenenfalls nur ein lokales Minimum ist, dessen zuverlässige Ermittlung nicht immer gewährleistet ist.

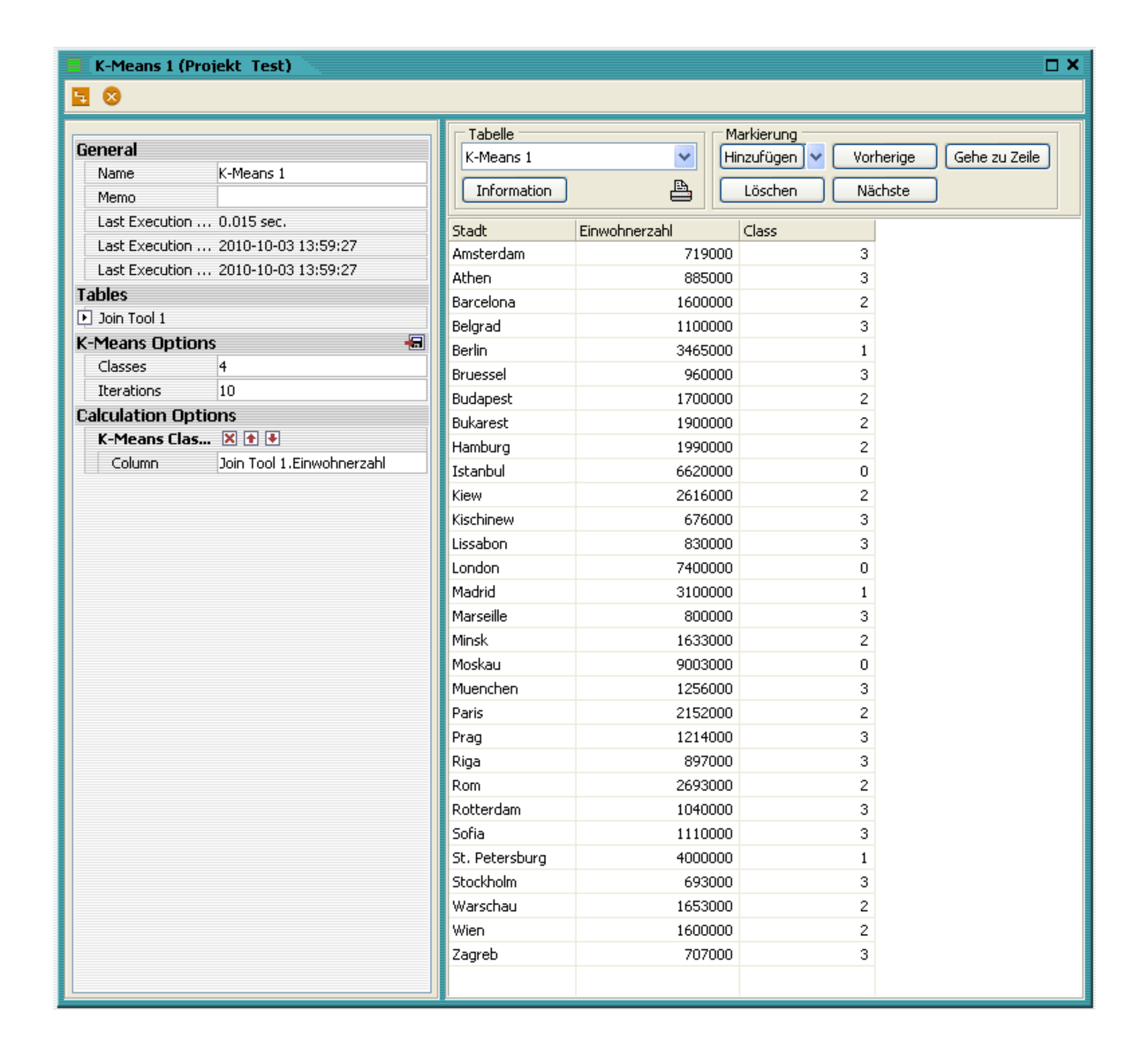
Tipp!
Es empfiehlt sich immer, mehrere K–Means–Durchgänge zur Überprüfung der Ergebnisstabilität von Gruppenbildungen durchzuführen, da die Ergebnisse mitunter stark von der zufälligen Ausgangslage der Gruppenzentren abhängen können.

Beispiel

Im Beispiel aus der Abbildung 4.15 (siehe Seite 88) sollen die Städte mit K–Means in vier Gruppe unterteilt werden. Die *Einwohnerzahl* dient dabei als Schlüsselgröße.

Um die Ergebnisse (siehe Abbildung 4.37) besser interpretieren zu können, zeigt die Abbildung 4.38 die Werte in sortierter Rei-

henfolge.



Stadt	Einwohnerzahl	Class
Amsterdam	719000	3
Athen	885000	3
Barcelona	1600000	2
Belgrad	1100000	3
Berlin	3465000	1
Bruessel	960000	3
Budapest	1700000	2
Bukarest	1900000	2
Hamburg	1990000	2
Istanbul	6620000	0
Kiew	2616000	2
Kischinew	676000	3
Lissabon	830000	3
London	7400000	0
Madrid	3100000	1
Marseille	800000	3
Minsk	1633000	2
Moskau	9003000	0
Muenchen	1256000	3
Paris	2152000	2
Prag	1214000	3
Riga	897000	3
Rom	2693000	2
Rotterdam	1040000	3
Sofia	1110000	3
St. Petersburg	4000000	1
Stockholm	693000	3
Warschau	1653000	2
Wien	1600000	2
Zagreb	707000	3

Abbildung 4.37.: Beispiel K–Means

Hinweis!
Abbildung 4.38 auf Seite 136 verdeutlicht, dass bei K–Means die Anzahl der Elemente in den einzelnen Gruppen deutlich differieren kann.

Stadt	Einwohnerzahl	Class
Moskau	9003000	0
London	7400000	0
Istanbul	6620000	0
St. Petersburg	4000000	1
Berlin	3465000	1
Madrid	3100000	1
Rom	2693000	2
Kiew	2616000	2
Paris	2152000	2
Hamburg	1990000	2
Bukarest	1900000	2
Budapest	1700000	2
Warschau	1653000	2
Minsk	1633000	2
Barcelona	1600000	2
Wien	1600000	2
Muenchen	1256000	3
Prag	1214000	3
Sofia	1110000	3
Belgrad	1100000	3
Rotterdam	1040000	3
Bruessel	960000	3
Riga	897000	3
Athen	885000	3
Lissabon	830000	3
Marseille	800000	3
Amsterdam	719000	3
Zagreb	707000	3
Stockholm	693000	3
Kischinew	676000	3

Abbildung 4.38.: Ergebnisse sortiert

4.3.12. Multiple Regression

Kurzbeschreibung	
Verwendungszweck	Lineare Regression mit mehreren unabhängigen Variablen.
Eingang	Eine Tabelle, eine Spalte numerischen Typs als abhänige, mindestens eine Spalte numerischen Typs als unabhängige Variable pro Funktionsblock
Ausgabe	Zwei Tabellen: 1. Eingangstabelle zzgl. je einer Ergebnisspalte pro Block 2. Tabelle mit einer Spalte Beschreibung und je einer Spalte Parameter pro Block
Besonderheit	-

Mit dem Multiple–Regressions–Modul wird eine lineare multiple Regression einer abhängiger Variable von mehreren unabhängigen Variable berechnet.

Definieren Sie die abhängige Variable unter *Dependent Column*. Anschließend können Sie mehrere unabhängige Variablen auswählen: Nachdem unter *Independent Column* eine Spalte eingetragen wurde, erscheint jeweils ein neues, leeres Eingabefeld am Ende der Liste. Tragen Sie auf diese Weise so viele Spalten ein, wie Sie für die Analyse benötigen.

Erzeugt werden zwei neue Tabellen:

1. Die Quelldaten zuzüglich der interpolierten Daten der jeweiligen Regressionsgeraden.

2. Enthält die berechneten Parameter der erzeugten Regressionen.

Hinweis!
Für einfache lineare oder quadratische Regressionen verwenden Sie bitte das Modul Regression (siehe 4.3.15, Seite 145).

4.3.13. Pivot

Kurzbeschreibung	
Verwendungszweck	Grupppierung und Aggregation in Kreuztabelle
Eingang	-
Ausgabe	nur grafische Ausgabe
Besonderheit	-

Dieses Modul berechnet und stellt eine Pivot–Ansicht zur Verfügung. Eine Pivot–Ansicht ist eine besondere Form einer Kreuztabelle, die zusätzlich Gruppierungen und Aggregate aus diesen definierten Gruppen anzeigt.

Die Berechnungen dieses Moduls hängen von der gewünschten Konfiguration der Ansicht ab. Bevor Daten angezeigt werden können, muss die Zeilen- und Spaltengruppierung gewählt und der Datenbereich konfiguriert werden. In Abbildung 4.39 sehen Sie ein Pivot–Modul im Ausgangszustand.

Gruppierung wählen

Ziehen Sie mit der Maus die gewünschten Tabellenspalten zur Gruppierung auf den Spalten- (1) oder Zeilengruppenbereich

(2) , sie wird anschließend als neues Gruppierungsfeld dargestellt.

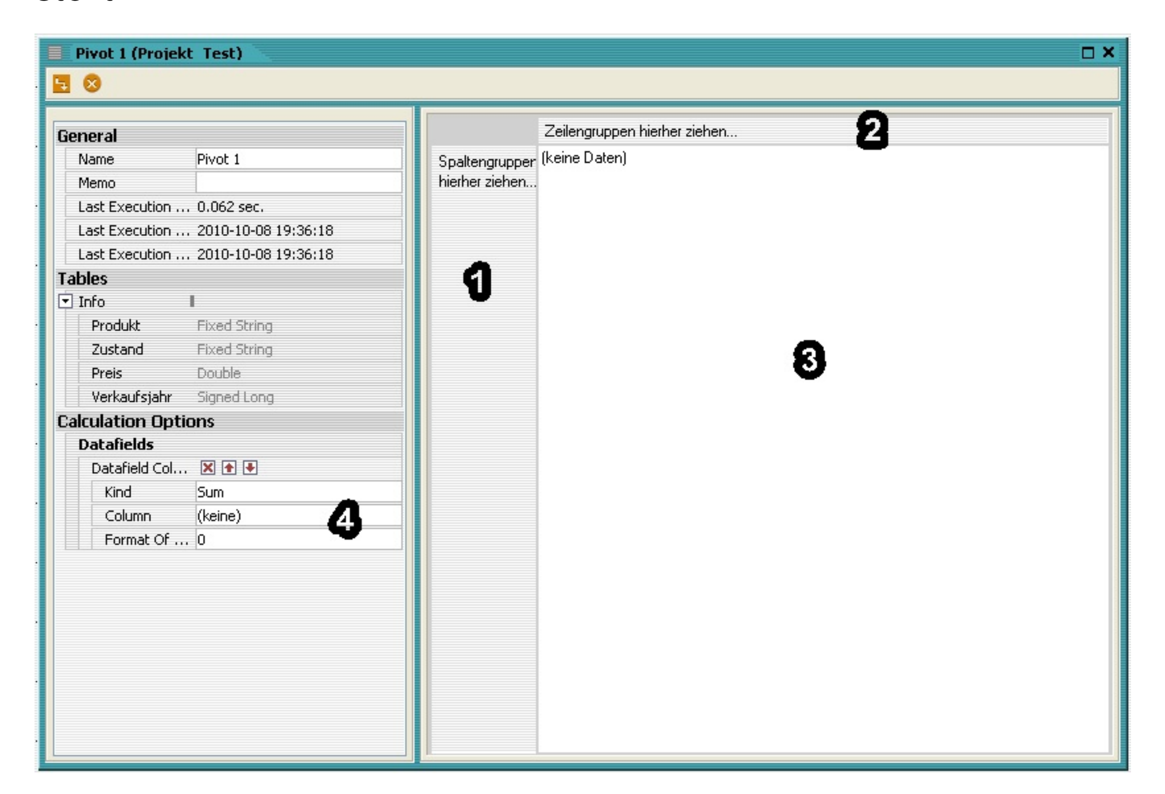


Abbildung 4.39.: Grundansicht Pivot

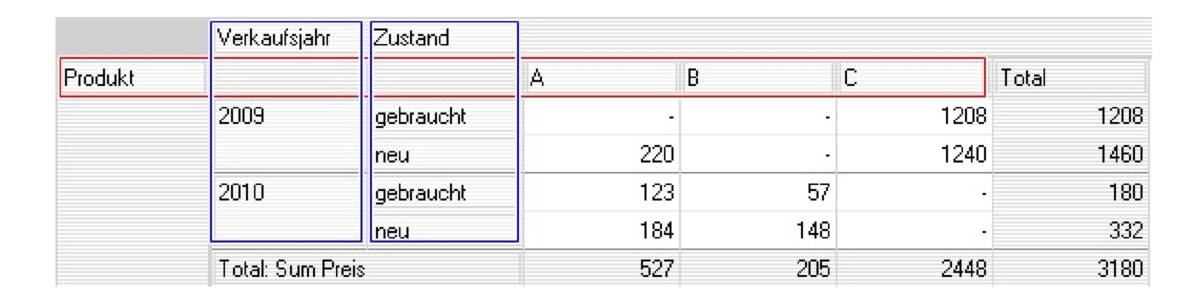
1. Spaltengruppen
2. Zeilengruppen
3. Datenbereich
4. Einstellungen des Datenbereichs

Um eine Spalte vor oder zwischen ein bestehendes Gruppierungsfeld einzufügen, ziehen Sie sie mit der Maus über die gewünschte Stelle. Über eingeblendete Einfügemarken wird die Stelle dargestellt, an der die neue Gruppierung eingefügt werden kann. Möchten Sie eine Spaltengruppierung durch eine andere ersetzen, ziehen Sie die gewünschte Spalte auf ein bestehendes Gruppierungsfeld. Beachten Sie bitte, dass jede Spalte nur einmal für eine Spalten- und Zeilengruppierung ausgewählt werden kann. Ziehen sie eine zu einer Gruppierung bereits verwendeten Spalte

auf ein neues Feld, wird sie am gewünschten Ort eingefügt und am bisherigen entfernt.

Die Reihenfolge der Felder gibt die jeweilige Ordnung der Gruppierung an. Einzelne Felder lassen sich aus der Gruppierung wieder löschen, indem Sie mit der rechten Maustaste im Kontext–Menü *Delete* wählen.

> **Hinweis!**
> Zeilengruppierungen werden im oberen Feld definiert, Spaltengruppierungen im linken. Die Reihenfolge und Anordnung entspricht den nach der ersten durchgeführten Berechnung angezeigten Gruppen und ist über kreuz zu lesen. Abbildung 4.40 zeigt, wie die Zuordnungen der einzelnen Gruppentitel zu verstehen ist.



	Verkaufsjahr	Zustand				
Produkt			A	B	C	Total
	2009	gebraucht	-	-	1208	1208
		neu	220	-	1240	1460
	2010	gebraucht	123	57	-	180
		neu	184	148	-	332
	Total: Sum Preis		527	205	2448	3180

Abbildung 4.40.: Gruppentitel

Aggregation bestimmen

Wählen Sie anschließend in einem Funktionsblock des Moduls (4) die Tabellenspalte und Aggregationsart, die in der Kreuztabelle angezeigt werden soll. Es lassen sich beliebig viele Aggregationen anlegen, bedenken Sie jedoch, dass die Pivot–Tabelle dabei schnell unübersichtlich werden kann.

Funktion	**Bezeichnung**
Count	Anzahl aller Einträge dieser Spalte
Count valid	Anzahl aller Einträge dieser Spalte ungleich NULL
Sum	Summe der Werte
Min	Kleinster Wert
Max	Größter Wert
Variance	Varianz der Werte
Std Deviation	Standardabweichung
Quad Sum	Quadratsumme
Quad Avg	Quadratdurchschnitt

Tabelle 4.15.: Aggregat–Funktionen Pivot

	Verkaufsjahr	Zustand				
Produkt			A	B	C	Total
	2009	gebraucht	-	-	1208	1208
		neu	220	-	1240	1460
	2010	gebraucht	123	57	-	180
		neu	184	148	-	332
	Total: Sum Preis		527	205	2448	3180

Abbildung 4.41.: Einfache Ergebnistabelle Pivot

1. Zeilengruppen
2. Spaltengruppen
3. Datenbereich
4. Zeilen–/Spaltensummen

Sobald Sie das Modul ausführen, werden die notwendigen Berechnungen ausgeführt und die Ergebnisse in der Tabellenansicht angezeigt. In Abbildung 4.41 sehen Sie eine einfache Ergebnistabelle mit einer Spalten- und zwei Zeilengruppierungen sowie einer Aggregation im Datenbereich.

Ist mehr als eine Aggregation angelegt, wird in der Pivot–Tabelle

zusätzlich eine Spalte mit den entsprechenden Bezeichnungen eingeblendet (Abbildung 4.42).

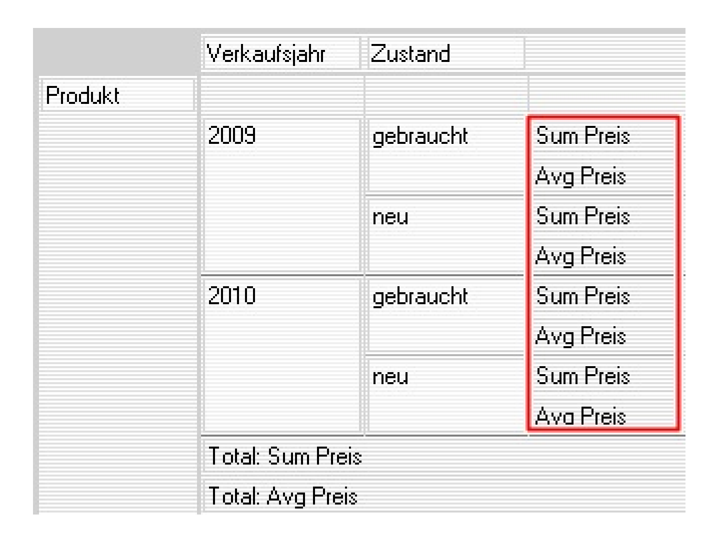


Abbildung 4.42.: Aggregationsbeschreibungen

Zwischensummen

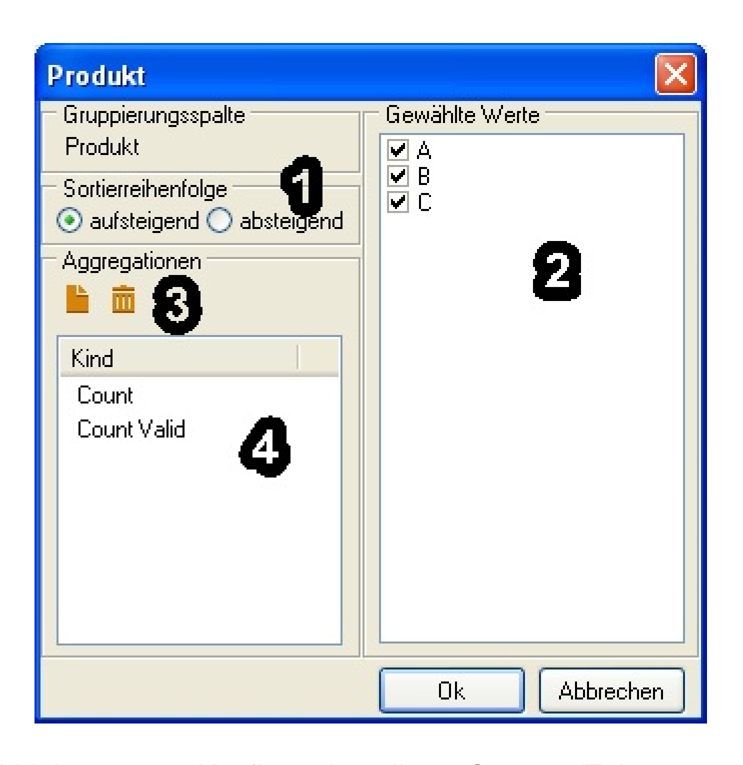


Abbildung 4.43.: Konfigurationsdialog Spalten-/Zeilengruppen

Doppelklicken Sie mit der Maus auf einen Eintrag der Zeilen- oder Spaltengruppenköpfe (Abbildung 4.39, (1) oder (2)) öffnet sich

der Dialog aus 4.43. Sie erhalten hier die Möglichkeit, die Anzeige der jeweiligen Spalten- oder Zeilengruppe zu konfigurieren. Mit der Auswahl (1) bestimmen Sie die Sortierreihenfolge, mit der Spalte (2) die anzuzeigenden Gruppen. Beachten Sie, dass diese Liste erst zur Verfügung steht, wenn die Pivot–Berechnung zuvor mindestens einmal ausgeführt wurde.

In der Aggregations–Liste (4) lassen sich Zwischensummen der jeweiligen Spalten-/Zeilengruppe definieren. Klicken Sie auf die entsprechende Schaltfläche in der Leiste (3), um eine neue Zeile einzufügen. Klicken Sie mit der Maus in der Spalte *Kind* auf einen Wert, können Sie zwischen den verschiedenen Aggegat–Funktionen (Tabelle 4.24) wählen.

	Verkaufsjahr	Zustand				
Produkt				A	B	C
	2009	gebraucht	Sum Preis	-	-	1208
			Avg Preis	-	-	402.667
		neu	Sum Preis	220	-	1240
			Avg Preis	110.000	-	620.000
		Sum Preis		220	-	2448
		Sum Preis		220.000	-	2448.000
	2010	gebraucht	Sum Preis	123	57	-
			Avg Preis	61.500	28.500	-
		neu	Sum Preis	184	148	-
			Avg Preis	92.000	49.333	-
		Sum Preis		307	205	-
		Sum Preis		307.000	205.000	-

Abbildung 4.44.: Aggregationsbeschreibungen

Nach dem Ausführen des Moduls erscheinen in der Tabelle für jede Aggregation (siehe 4.3.13, Seite 140) zusätzlich die gewünschten Zwischensummen, die aus den darüber liegenden Werten errechnet sind (Abbildung 4.44).

Spaltenbreite verändern

Klicken Sie mit der Maus in die Abgrenzung zwischen dem Spaltengruppentitel (Abbildung 4.40, (1) und dem Datenbereich (2)), um die Breite aller Spalten gleichzeitig, oder zwischen zwei Zeilengruppen am oberen Fensterrand, um eine einzelne Spalte zu verändern.

4.3.14. Quantiles

Kurzbeschreibung	
Verwendungszweck	Berechnung der empirischen und theoretischen Quantile
Eingang	Eine Tabelle, je eine Spalte numerischen Typs pro Funktionsblock
Ausgabe	Zwei Tabellen: 1. Eingangstabelle 2. Tabelle mit einer Spalte Beschreibung der Schrittweite und je einer Spalte Ergebnisse pro Block
Besonderheit	-

Berechnet die empirischen bzw. theoretischen (einer Verteilung zugrunde liegenden) α–Quantilen der Spalten einer Tabelle. Die Parameter *Min* und *Max* legen den niedrigsten und hochsten α–Wert fest, *Step* die Intervallbreite.

Bezeichnung	**Werte**
Kind	Bestimmt die Art der Berechnung der Quantilen. *Empirical Quantiles* sind die empirischen Quantilen der jeweiligen Quellspalte. *Normal Distributed Quantiles* sind die gemäß den Untereinstellungen definierten theoretischen Quantilen der Normalverteilung.

Auto	Ist diese Option gewählt, werden die Werte der Quellspalte zur Berechnung verwendet, ansonsten die angegebenen festen Wert.
Mean (Nur „Normal“)	Gibt den zu verwendenden Mittelwert an.
Standard Deviation (Nur „Normal“)	Gibt die zu verwendende Standardabweichung an.
Decay Factor (Nur „Exponential“)	Gibt die Ausfallrate an.
Minimum (Nur „Exponential“)	Gibt die Verschiebnung des Null–Punktes der Verteilung an.
Column	Spalte, für die die Quantile berechnet werden sollen
Name	Name der zu erzeugenden Spalte

Tabelle 4.16.: Parameter Quantiles

4.3.15. Regression

Kurzbeschreibung	
Verwendungszweck	Berechnung der linearen oder quadratischen Regression
Eingang	Eine Tabelle, je eine Spalte als unabhängige und abhängige Variable pro Funktionsblock
Ausgabe	Zwei Tabellen: 1. Eingangstabelle zzgl. einer Ergebnisspalte pro Block 2. Tabelle mit einer Spalte Beschreibung und je einer Spalte Parameter pro Block
Besonderheit	-

Das Modul führt gemäß Ihrer Auswahl eine lineare *(Linear)* oder quadratische *(Quadratic)* Regression anhand der ausgewählten Spalten durch. Für eine Berechnung sind zwei Spalten als Eingangsdaten notwendig: die *Independent Column* und die *Dependent Column*.

Erzeugt werden zwei neue Tabellen:

1. Die Quelldaten zuzüglich der interpolierten Daten der jeweiligen Regressionsgeraden.
2. Enthält die berechneten Parameter der erzeugten Regressionen.

Anwendungsgebiet

Besteht eine begründete Vermutung, dass eine bestimmte Variable A auf eine Variable B einen bestimmten Einfluss hat (bspw. die durchschnittliche Tagestemperatur auf den Umsatz eines bestimmten Konsumartikels) bzw. die Richtung des Zusammenhangs fest steht, kann aufgrund der Ausprägung von A B prognostiziert werden.

Diese Prognose setzt voraus, dass entweder ein linearer Zusammenhang zwischen den Variablen vorliegt oder dass ein Zusammenhang besteht, der keine Linie, sondern einen Bogen beschreibt. Ein solcher Verlauf wäre beispielsweise gegeben, wenn man Wachstumsprozesse beschreiben möchte, die anfangs langsam, später aber sprunghaft ansteigen. Möchte man zum Beispiel die Ausbreitung einer Schädlingsart prognostizieren, kann man sich leicht vor Augen führen, dass die Vermehrung einer Tierart immer schneller verläuft, je mehr davon in fortpflanzungsfähigem Alter existieren. Je stärker dieser Anteil fortpflanzungsfähiger Tiere im Verhältnis zur Sterberate wächst, umso sprunghafter ist das Wachstum dieser Population.

Allerdings sollte man sich immer vor Augen halten, dass quadratisches Wachstum in den allermeisten Fällen nur für eine gewisse

Zeit existieren kann, da begrenzende Ressourcen, wie Nahrungsangebot oder Lebensraum dem Wachstum ein Ende setzen werden; so wie der einstige quadratisch anwachsende Umsatz von Mobilfunkgeräten oder die Nutzung von E–Mail im Laufe der Zeit nur so lange vorherrschen konnte, bis ein begrenzend hohes Maß an Käufern bzw. Nutzern erreicht wurde.

Wachstumsfunktionen sind ihrer Natur nach eigentlich exponentiell. Das bedeutet, der Exponent wächst im Verlauf der Zeit, während er beim quadratischen Modell konstant gleich 2 ist. Durch die natürliche Begrenzung des exponentiellen Wachstums lassen sich viele Wachstumsprognosen quadratisch jedoch gut schätzen, auch wenn diese Schätzung meist etwas konservativ ausfällt.

Bedienung

Zu Beginn einer solchen Modellerstellung wählen Sie unter *Kind* die Art des erwarteten Zusammenhangs aus, ob linear (proportional) oder quadratisch (bogenförmig). Wählen Sie anschließend die Spalten aus, die die folgenden Variablen enthalten: Die abhängige *(dependent)* Variable ist die zu erklärende oder zu prognostizierende Variable. Die unabhängige *(independent)* Variable ist die erklärende oder Prognosevariable.

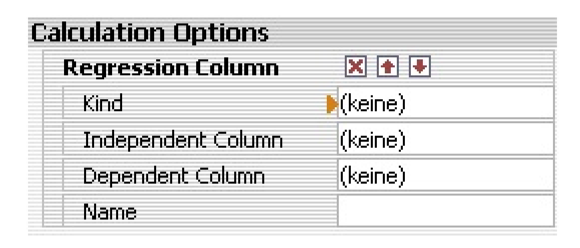


Abbildung 4.45.: Modul Regression: Calculation Options

Ergebnisinterpretation

Als Ergebnis wird eine Tabelle mit der Modellfunktion ausgegeben (gekennzeichnet durch das Suffix „Par“), die die abhängige Variable optimal anhand der unabhängigen Variable prognostiziert (un-

ter der Annahme eines linearen oder quadratischen Zusammenhangs). Die Tabelle enthält die errechneten Werte der folgenden Funktionen:

Linear Regression:

$$Y = Gradient \cdot X + Intercept$$

wobei Y für die *dependent* Variable und X für die *independent* Variable steht.

Quadratisches Modell:

$$Y = b_2 \cdot X^2 + b_1 \cdot X + a$$

wobei Y für die *dependent* Variable und X für die *independent* Variable steht.

Der *Mean Square Error* ist die durchschnittliche quadrierte Abweichung eines prognostizierten Wertes vom tatsächlichen Wert, also ein Maß für die Genauigkeit der Vorhersagen.

Obwohl in der quadratischen Modellbeschreibung der Fehleranteil immer geringer ausfallen wird als im linearen Modell (weil das quadratische Modell das lineare mit beinhaltet), sollte man dieses nur übernehmen, wenn es eine sinnvolle inhaltliche Erklärung dafür gibt. Im Zweifel, bei ähnlich hohem Fehleranteil, empfiehlt sich das sparsamere lineare Modell.

Ein recht deutliches Maß für die Prognosequalität ist die Varianzaufklärung des Modells, die sich durch die Quadrierung der Korrelation (r^2) zwischen prognostizierten und realisierten Werten der abhängigen Variable errechnet.

Aufklärungen von vier Prozent ($r^2 = 0,04$) werden oft schon bei vielen Messreihen interpretiert und können auf Effekte hindeuten, während $r^2 = 0,1$ schon auf deutliche Zusammenhänge schließen lässt.

Es ist ein statistisches Artefakt, dass, je mehr Variablen man zu einem Modell hinzunimmt, umso mehr Varianz aufgeklärt wird. Es ist unmittelbar einsichtig, dass ein Modell mit zehn erklärenden Variablen mit einer Varianzaufklärung von 4$\%$ wenig Sinn macht.

Um den Zuwachs an Varianz durch das bloße Hinzunehmen neuer Variablen herauszurechnen, wird das adjustierte R^2 berechnet (*Adjusted R Square*), das die korrigierte Varianzaufklärung wiedergibt.

Beispiel

Untersuchen wir nun, ob zwischen den Variablen X (Temperatur) und Y (Sonnenstunden) (siehe Abbildung 4.46) eine lineare Beziehung existiert.

Temperatur (in Grad Celsius)	Anzahl Sonnenstunden
15,1	5
18,5	6
20,5	7
23	7
25,7	8
26,3	9,5
28,1	9
21,9	7
22,4	7,5
30,2	8
17,5	6
22	5,5
27,5	10

Abbildung 4.46.: Beispieltabelle für Regressionsanalyse

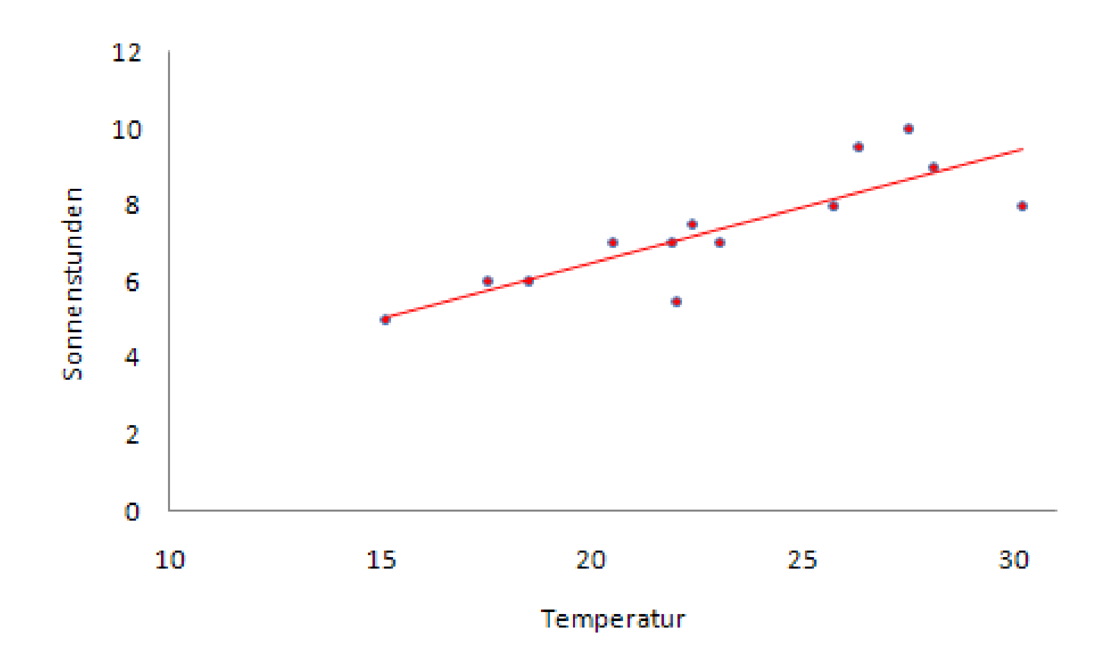
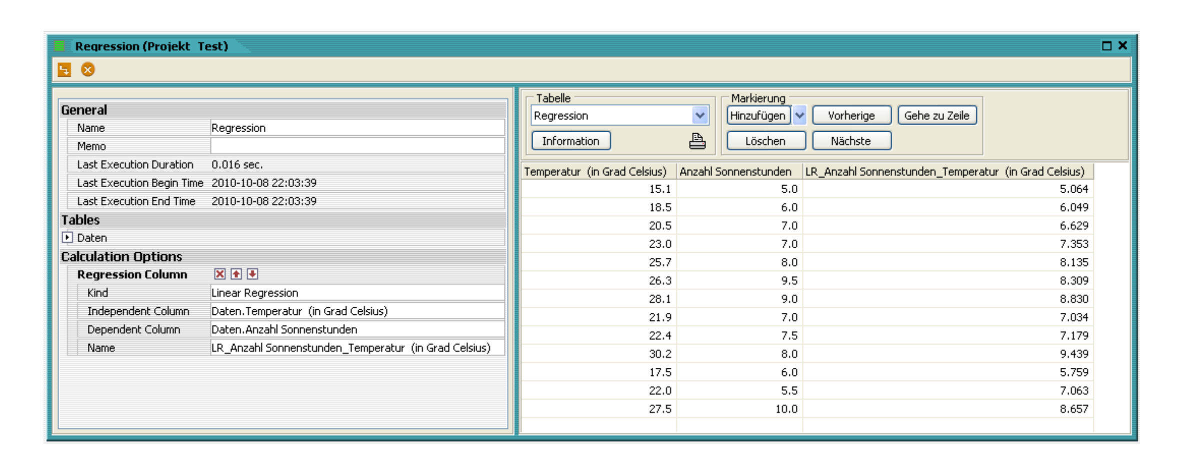


Abbildung 4.47.: Graphische Darstellung der Daten der Abbildung 4.46

Das Modul Regression wird wie folgt (Abbildung 4.48) eingestellt:

- Kind: *Linear Regression*
- Independent Column: *Temperatur*
- Dependent Column: *Anzahl Sonnenstunden*



Temperatur (in Grad Celsius)	Anzahl Sonnenstunden	LR_Anzahl Sonnenstunden_Temperatur (in Grad Celsius)
15.1	5.0	5.064
18.5	6.0	6.049
20.5	7.0	6.629
23.0	7.0	7.353
25.7	8.0	8.135
26.3	9.5	8.309
28.1	9.0	8.830
21.9	7.0	7.034
22.4	7.5	7.179
30.2	8.0	9.439
17.5	6.0	5.759
22.0	5.5	7.063
27.5	10.0	8.657

Abbildung 4.48.: Einstellungen und Ergebnisse der Regression

Die Tabelle *Regression* (siehe Abbildung 4.48) erhält eine neue

Spalte, die die prognostizierten Werte enthält.

Die relevanten statistischen Werte der Regression lassen sich in der Tabelle *Regression_Par* (Abbildung 4.49) einsehen.

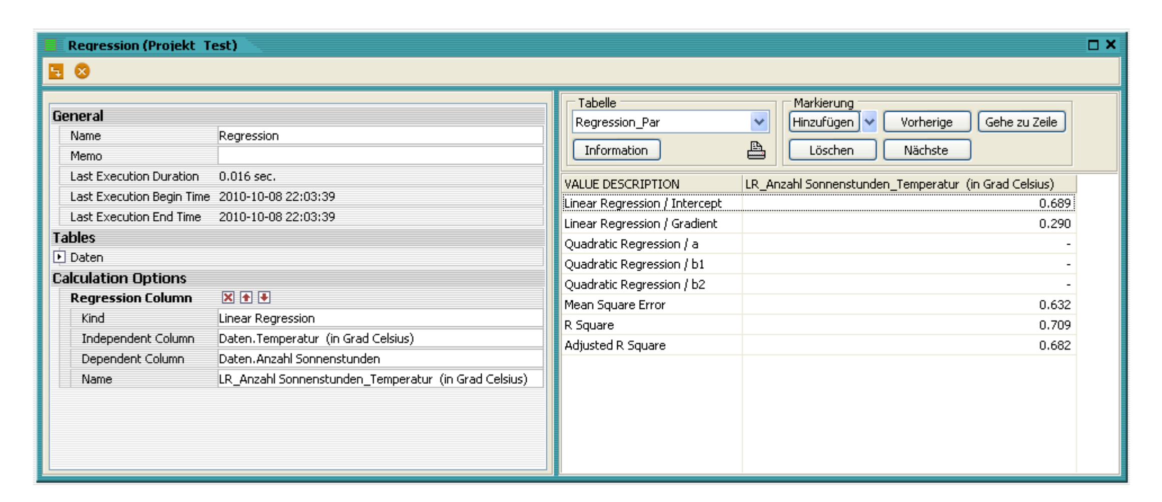


Abbildung 4.49.: Statistische Werte der Regression

Die lineare Funktion lautet

$$Anzahl\ Sonnenstaunden = 0,29 \cdot Temperatur + 0,689 \quad .$$

Das Bestimmtheitsmaß R^2 (*R Square*) ist 0,709 bei einem *Mean Square Error* von 0,632. Das adjustierte Bestimmtheitsmaß liegt bei 0,682.

Interpretation Bestimmtheitsmaß

Das Bestimmtheitsmaß R^2 mißt den Zusammenhang zwischen der abhängigen Variable Y und den unabhängigen Variablen $(X_1, X_2, \ldots, X_n)$.
Der Wertebereich von R^2 liegt zwischen
0 (= kein Zusammenhang zwischen Y und $(X_1, X_2, \ldots, X_n)$
und
1 (= perfekter linearer Zusammenhang zwischen Y und $(X_1, X_2, \ldots, X_n)$.

Hinweis!
Falls der $Gradient = 0$ ist, so kann R^2 negative Werte annehmen.

Hinweis!
Je näher R^2 sich 1, desto besser erklärt die unabhängige Variable X die abhängige Variable Y. Bei $R^2 = 1$ liegen alle Datenpunkte auf der Regressionsfunktion.

Das Bestimmtsheitsmaß hat einen Nachteil: Je mehr unabhängige Variablen $(X_1, X_2, \ldots, X_n)$ eine abhängige Variable Y beschreiben, desto größer wird das Bestimmtheitsmaß. Dabei spielt es keine Rolle, ob die neu eingefügten unabhängigen Variablen eine Relevanz haben.

Das adjustierte Bestimmtheitsmaß $\overline{R}^2$ berücksichtigt die Anzahl der unabhängigen Variablen und ist bei mehreren unabhängigen Variablen dem Bestimmtsheitsmaß R^2 vorzuziehen.

4.3.16. SimAnalysis

Kurzbeschreibung	
Verwendungszweck	Vergleicht eine Textspalte mit einer Liste von Strings und gibt Informationen zur Ähnlichkeit von Einträgen mit Werten der Textspalte aus.
Eingang	Eine Tabelle, eine Textspalte mit Werten und eine Stringspalte mit den Vergleichsstrings
Ausgabe	Zwei Tabellen: 1. Eingangstabelle mit der Textspalte, evtl. erweitert um einige Spalten mit den ähnlichsten Strings und ihrem Ahnlichkeitswert 2. Eingangstabelle mit der Vergleichsspalte, evtl. zu einigen Ähnlichkeitsleveln die Anzahl der Vorkommen in der Textspalte sowie evtl. Beispiele zu den angegebenen Ähnlichkeitsleveln
Besonderheit	-

Das Modul sucht zu einem String die ähnlichsten aus einer Vergleichsliste und gibt diese mit einem aus. Gleichzeitig wird zu jedem Vergleichswert die Anzahl der Treffer zu bestimmten Ähnlichkeitsniveaus bestimmt. Die genauen Werte hängen von den Parametern ab.

Funktion	Bezeichnung
Search Column	Spalte mit den Vergleichswerten
Method of Similarity Calculation	Auswahl des Algorithmus zur Ähnlichkeitssuche: Levenstein und Fuzzy Ngram

Case Sensitive	Berücksichtigung der Groß- und Kleinschreibung
Generate Similarity Matrix	Ausgabe der ähnlichsten Strings
Number of Similar Token to show	Anzahl der zu zeigenden Strings
Minimum Similarity for Match	Minimale Ähnlichkeit des Vergleichsstrings um angezeigt zu werden Wertebereich: $(0; 1)$
Generate Similarity Information	Anzeige der Vergleichsspalte der Trefferhäufigkeiten zu jedem Wert
Number of Similarity	Anzahl der Ähnlichkeitsniveaus, zu denen Häufigkeiten ausgegeben werden sollen
Minimum Similarity	Niedrigstes Ähnlichkeitsniveau zu dem eine Treferhäufigkeit ausgegeben werden soll Wertebereich: $(0; 1)$
Show Samples for Similarity Levels	Anzeige eines Beispiels aus dem Ähnlichkeitsbereich

Tabelle 4.17.: SimAnalysis: Similarity Options

Beispiel

Name	Name_2
Schuettler	Schuetler
Becker	Schuller
Stich	Bekker
Kiefer	Bäcker
	Stier
	Stiech
	Kieffar
	Kiefer

Abbildung 4.50.: Beispiel SimAnalysis

Es sollen Namen (Abbildung 4.50) miteinander verglichen werden.

Mit Levenstein wird eine Ähnlichkeitssuche gestartet.

Die Tabelle *SimAnalysis 2* (Abbildung 4.51) liefert jedem String der Spalte *Name*_2 einen Vorschlag in der Spalte *Token 0* mit einem Ähnlichkeitsmaß in der Spalte *Token 0 Similarity*.

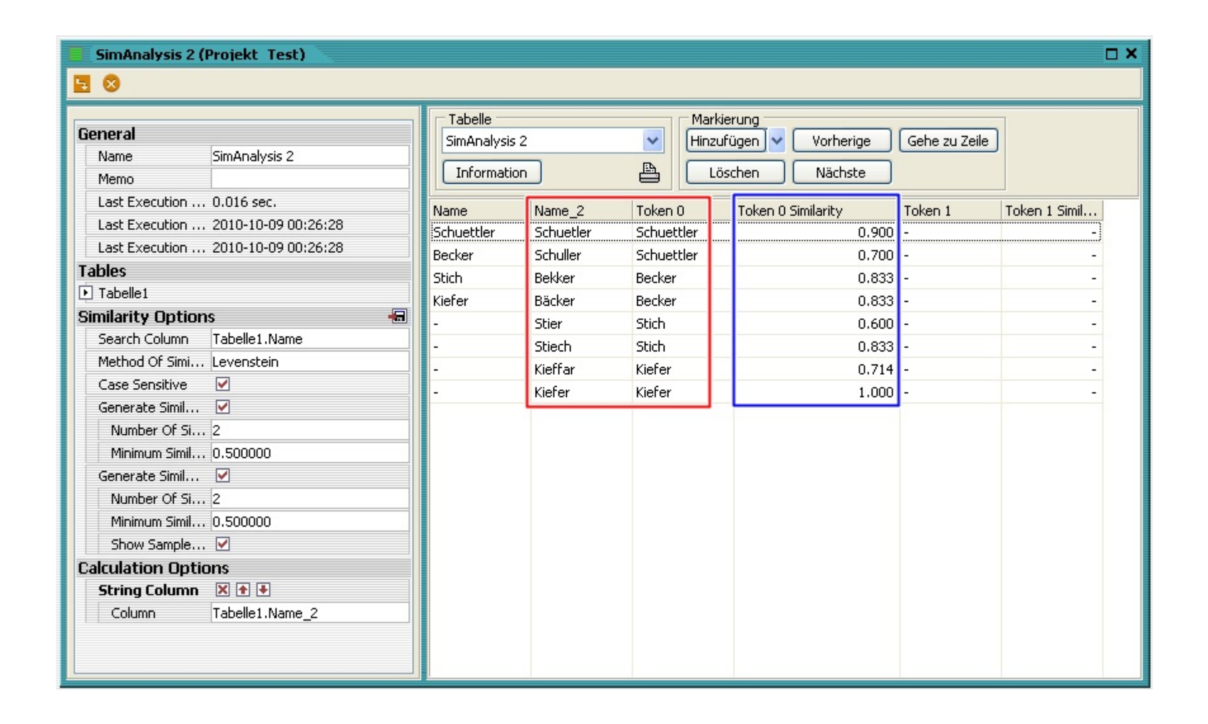


Abbildung 4.51.: SimAnalysis: Levenstein

Die Tabelle *SimAnalysis 2_Par* (Abbildung 4.52) gibt eine Übersicht über die Anzahl der Ähnlichkeitstreffer von *Name* in *Name_2* in dem jeweiligen Ähnlichkeitslevel. Die Spalten *Sample String* bzw. *Sample String_1* zeigen jeweils ein Beipiel.

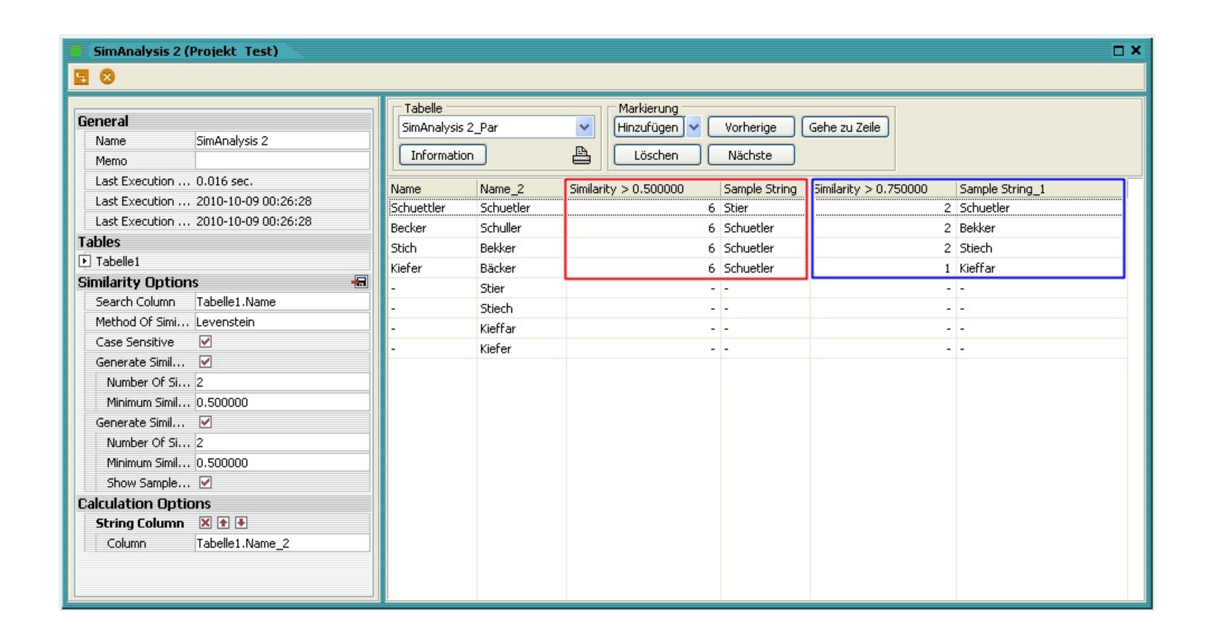


Abbildung 4.52.: SimAnalysis_Par: Levenstein

4.3.17. Sort

Kurzbeschreibung	
Verwendungszweck	Sortiert eine Tabelle
Eingang	Eine Tabelle, beliebig viele Spalten
Ausgabe	Eingangstabelle in sortierter Reihenfolge
Besonderheit	-

Das Modul erzeugt eine neue Tabelle mit den sortierten Spalten der ausgewählten Quelldaten, sowie eine weitere Spalte mit den Häufigkeiten der sortierten Werte.

Legen Sie mehrere Sortierungs–Blöcke in den *Calculation Options* an, wird entsprechend deren Reihenfolge eine Untersortierung durchgeführt. Das bedeutet, dass bei gleichen Werten die jeweils niedrigere Sortierung über die Reihenfolge entscheidet.

Bezeichnung	**Werte**
Column	Spalte, nach der sortiert werden soll
Sort Oder	Sortierreihenfolge *Ascending:* aufsteigend *Descending:* absteigend

Tabelle 4.18.: Parameter Sort

4.3.18. Standard Quantiles

Kurzbeschreibung	
Verwendungszweck	Berechnet die gebräuchlichsten Quantile
Eingang	Eine Tabelle, je eine numerische Spalte pro Funktionsblock
Ausgabe	Zwei Tabellen: 1. Eingangstabelle 2. Tabelle mit einer Spalte Beschreibung und je einer Ergebnisspalte pro Block
Besonderheit	-

Übernimmt die Quelltabelle und erzeugt zusätzlich eine Tabelle, in der die wichtigsten α–Quantilen (Abbildung 4.53) der in den Funktionsblöcken ausgewählten Spalten enthalten sind.

Bezeichnung	**Werte**
Column	Spalte, für die die Quantile berechnet werden sollen
Name	Name der zu erzeugenden Spalte

Tabelle 4.19.: Parameter Standard Quantiles

Value Description	Std_Quantiles_Temperatur (in Grad Celsius)
Median	15.100
1-Quartile	27.500
2-Quartile	15.100
3-Quartile	17.500
5-Percentile	23.000
10-Percentile	26.300
20-Percentile	30.200
30-Percentile	27.500
40-Percentile	15.100
50-Percentile	15.100
60-Percentile	15.100
70-Percentile	17.500
80-Percentile	18.500
90-Percentile	20.500
95-Percentile	21.900

Abbildung 4.53.: Berechnete Standard–Quantile

4.3.19. Tokenizer

Kurzbeschreibung	
Verwendungszweck	Erzeugt aus einer oder mehreren Textspalten eine Liste von Token (Substrings) die nach vorgegebenen Regeln gebildet werden.
Eingang	Eine Tabelle, eine oder mehrere Textspalten
Ausgabe	Zwei Tabellen: 1. Eingangstabelle, evtl. mit einigen Spalten mit Informationen zu den verwendeten Token 2. Tabelle mit einer Spalte, in der die vorkommenden Token stehen, evtl. zusätzliche Informationen zum Vorkommen der einzelnen Token.
Besonderheit	-

Dieses Modul bildet aus den Strings einer oder mehrerer Spalten (wobei diese Spalten auch Zahlen oder Datumsspalten sein können) eine Liste von *Token* (Substrings) die in einer neuen Tabelle gespeichert werden. Unter *Calculation Options* wählen sie die Spalten aus, aus denen die Token gebildet werden. Unter *Tokenizer Options* (vgl. Tabelle 4.21) bestimmen sie, wie die Substrings gebildet werden, und welche Zusätzlichen Informationen hierzu ausgegeben werden.

Funktion	**Bezeichnung**
Method of Token Generation	Trennzeichen(Delimiter) oder fester Länge(Fixed Token Length)
Delimiters (nur Delimiters)	String mit den Zeichen, die ein Tokenende markieren
Length (nur Fixed Length)	Länge der Token
Case Sensitive	Berücksichtigung der Groß- und Kleinschreibung
Only Token with frequency >	Minimale Häufigkeit zur Aufnahme des Tokens in die Tabelle
Only Token with frequency <	Maximale Häufigkeit zur Aufnahme des Tokens in die Tabelle (-1 bedeutet die Übernahme aller Token)
Method of Token Selection	Eine weitere Möglichkeit die Auswahl der Token einzuschränken: Alle Token oder nur die häufigsten bzw. die seltensten Token nur (least und most frequent).
Count	Anzahl der zu berücksichtigenden Token

Generate Token Information	Zu jedem Token erfolgt die Ausgabe von: Anzahl des Vorkommen, relative Häufigkeit des Vorkommens, minimale, maximale und durchschnittliche Häufigkeit in einer Zeile, Anzahl der Zeilen in denen es vorkommt, relative Anzahl der Zeilen in denen es vorkommt ausgegeben.
Generate Token Matrix	Ausgabe zusätzlicher Werte zu jeder Zeile zu einigen Token
Number of Token	Anzahl der Token zu denen zusätzliche Informationen ausgegeben werden
Method of Token Selection	Berücksichtigung der häufigsten oder seltensten Token
Choice of Value	Auswahl der ausgegebenen Information

Tabelle 4.20.: Tokenizer: Tokenizer Options

Beispiel

Zu den Einträgen in der Tabelle 4.54 wird das Modul Tokenizer (Abbildung 4.55) angewandt.

Spalte
A01;A02;A03
A01;A03
A02;A03
A01;A02;A04
A03;A04

Abbildung 4.54.: Beispiel Tokenizer

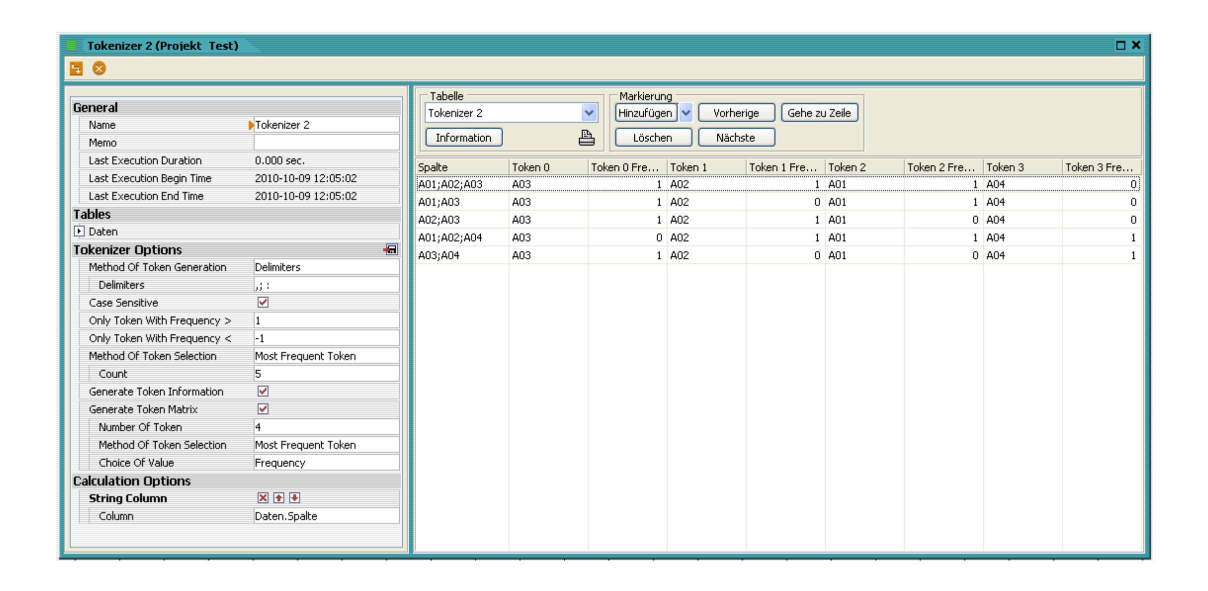
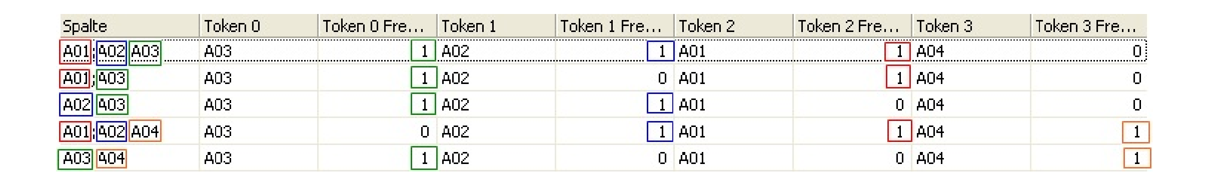


Abbildung 4.55.: Beispiel Tokenizer–Modul



Spalte	Token 0	Token 0 Fre...	Token 1	Token 1 Fre...	Token 2	Token 2 Fre...	Token 3	Token 3 Fre...
A01;A02;A03	A03	1	A02	1	A01	1	A04	0
A01;A03	A03	1	A02	0	A01	1	A04	0
A02;A03	A03	1	A02	1	A01	0	A04	0
A01;A02;A04	A03	0	A02	1	A01	1	A04	1
A03;A04	A03	1	A02	0	A01	0	A04	1

Abbildung 4.56.: Tokenizer: Beziehung zu den Token

Token	Token Count	Token Relati...	Token Minimum Count	Token Maximum Count	Token Average Count	Token Line C...	Token Relati...
A03	4	1.000	1	1	1.000	4	1.000
A02	3	0.750	1	1	1.000	3	0.750
A01	3	0.750	1	1	1.000	3	0.750
A04	2	0.500	1	1	1.000	2	0.500

Abbildung 4.57.: Tokenizer: Statistik der Token

4.3.20. Tokenmatrix

Kurzbeschreibung	
Verwendungszweck	Erzeugt aus einer oder mehreren Textspalten eine Liste von Token (Substrings) die nach vorgegebenen Regeln gebildet werden.
Eingang	Eine Tabelle, eine oder mehrere Textspalten
Ausgabe	Zwei Tabellen: 1. Eingangstabelle, evtl. mit einigen Spalten mit Informationen zu den verwendeten Token 2. Tabelle mit einer Spalte, in der die vorkommenden Token stehen, evtl. zusätzliche Informationen zum Vorkommen der einzelnen Token.
Besonderheit	Ergänzung zum Tokenizer

Das Tokenmatrix–Modul ist eine Ergänzung zum Tokenizer–Modul. Im Gegensatz zu diesem erzeugt es keine Tokenliste, sondern vergleicht die Wertespalten mit einer vorgegebenen Tokenliste, die auch aus einer anderen Quelle stammen kann. Hierzu wird zusätzlich eine Tokenspalte ausgewählt. Ansonsten sind die Attribute und generierten Spalten im Großen und Ganzen gleich. Die Bedeutung der *Token generation Options* bezieht sich hier darauf, wie die Teilstrings, die mit der Tokenliste verglichen werden, aus den Wertespalten gebildet werden.

Funktion	Bezeichnung
Method of Token Generation	Trennzeichen(Delimiter) oder fester Länge(Fixed Token Length)
Delimiters (nur Delimiters)	String mit den Zeichen, die ein Tokenende markieren
Length (nur Fixed Length)	Länge der Token
Case Sensitive	Berücksichtigung der Groß- und Kleinschreibung
Token Column	Spalte in der die Tokenliste steht
Number of Token	Anzahl der Token zu denen zusätzliche Informationen ausgegeben werden
Method of Token Selection	Berücksichtigung der häufigsten oder der seltensten Token.
Choice of Value	Ausgabe der Information: Häufigkeit, relative Häufigkeit oder generelles Vorkommen in dieser Zeile
Generate Token Information	Zu jedem Token erfolgt die Ausgabe von: Anzahl des Vorkommen, relative Häufigkeit des Vorkommens, minimale, maximale und durchschnittliche Häufigkeit in einer Zeile, Anzahl der Zeilen in denen es vorkommt, relative Anzahl der Zeilen in denen es vorkommt ausgegeben.
Generate Token Matrix	Ausgabe zusätzlicher Werte zu jeder Zeile zu einigen Token

Tabelle 4.21.: Tokenizer: Tokenizer Options

Beispiel

Zu den Einträgen in der Tabelle 4.58 wird das Modul Tokenizer (Abbildung 4.59) angewandt.

Spalte
A01;A02;A03
A01;A03
A02;A03
A03;A04
A01
A03

Abbildung 4.58.: Beispieltabelle für die Tokenmatrix

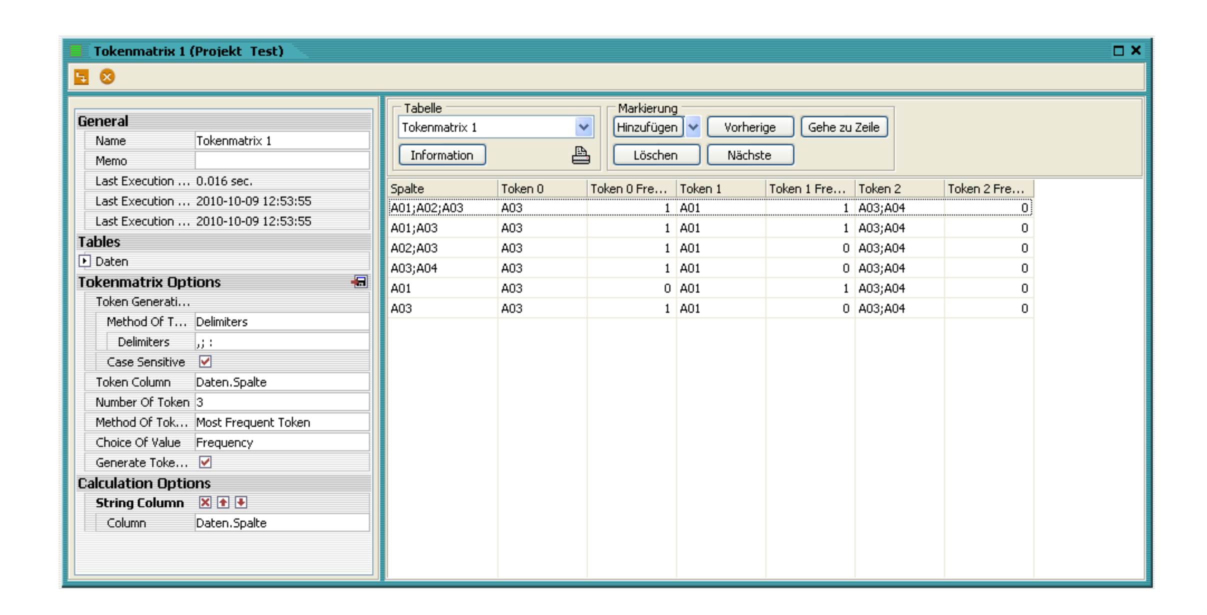


Abbildung 4.59.: Beispiel Tokenmatrix

Spalte	Token 0	Token 0 Fre...	Token 1	Token 1 Fre...	Token 2	Token 2 Fre...
A01;A02;A03	A03	1	A01	1	A03;A04	0
A01;A03	A03	1	A01	1	A03;A04	0
A02;A03	A03	1	A01	0	A03;A04	0
A03;A04	A03	1	A01	0	A03;A04	0
A01	A03	0	A01	1	A03;A04	0
A03	A03	1	A01	0	A03;A04	0

Abbildung 4.60.: Tokenmatrix: Beziehung zu den Token

Spalte	Token Count	Token Relati...	Token Minim...	Token Maxi...	Token Avera...	Token Line C...	Token Relati...
A01;A02;A03	0	0.000	0	0	-1.#IO	0	0.000
A01;A03	0	0.000	0	0	-1.#IO	0	0.000
A02;A03	0	0.000	0	0	-1.#IO	0	0.000
A03;A04	0	0.000	0	0	-1.#IO	0	0.000
A01	3	0.600	1	1	1.000	3	0.600
A03	5	1.000	1	1	1.000	5	1.000

Abbildung 4.61.: Tokenmatrix: Statistik der Token

4.3.21. Transformation

Kurzbeschreibung	
Verwendungszweck	Transformiert Werte einer Spalte
Eingang	Eine Tabelle, je eine numerische Spalte pro Funktionsblock
Ausgabe	Eingangstabelle zzgl. je einer Ergebnisspalte pro Funktionsblock
Besonderheit	-

Transformationen sind in der Regel dazu gedacht, Daten in einen für das jeweilige Einsatzgebiet tauglichen Wertebereich oder eine entsprechende Verteilung zu bringen. Damit können unter anderem Fehler durch unterschiedliche Skalierungen der Eingangsdaten verhindert oder gewünschte Verschiebungen der Werte erreicht werden.

In den folgenden Formeln ist mit x immer der jeweilige Quellwert bezeichnet, $min\left(x\right)$ beschreibt das Minimum, $max\left(x\right)$ das Maximum und N die Gesamtzahl der gültigen Werte der Spalte.

Ist die mit angegebene Nebenbedingung nicht erfüllt, so wird eine leere Ergebnismenge zurückgegeben.

Linear Transformation

Transformiert die Werte innerhalb eines wählbaren Bereichs über die Parameter $Trans_{min}$ und $Trans_{max}$:

$$x_{Trans} = Trans_{min} + (Trans_{max} - Trans_{min}) \cdot \frac{x - min\,(x)}{max\,(x) - min\,(x)}$$

wobei

- $Trans_{min} < Trans_{max}$ und
- $min\,(x) < max\,(x)$

gelten muss.

Reciprocal Transformation

Berechnet den Kehrwert gemäß der Formel:

$$x_{Trans} = \frac{1}{x}$$

wobei $x \neq 0$ sein muss.

Sqrt Transformation

Berechnet die Wurzeltransformation der Werte gemäß der Formel:

$$x_{Trans} = \sqrt{x - \delta}$$

wobei $\delta \leq min\,(x)$ sein muss. Mit dem Parameter δ kann eine Verschiebung der Kurve erreicht werden.

Logarithmic Transformation

Berechnet den natürlichen Logarithmus gemäß der Formel:

$$ln\,(x - min\,(x))$$

Für den kleinsten Wert x der Eingangsspalte lautet das Ergebis stets NULL, da $x_{Trans} = ln\,(0)$ nicht definiert ist.

Box Cox Transformation

Transformiert Zeitreihen nach dem Box–Cox–Verfahren gemäß der Formel:

$$x_{Trans} = \frac{(x-\delta)^{\lambda} - 1}{\lambda}$$

wobei $\lambda \neq 0$ sein muss.

Arc Sine Transformation

Bringt geeignete Quelldaten annähernd auf eine Normalverteilung, gemäß der Formel:

$$x_{Trans} = \sqrt{w_1 + N} \cdot arcsin\left(\sqrt{\frac{x + w_2}{N + w_3}}\right)$$

wobei $N + w_3 > 0$ sein muss.
Über die Parameter w_1, w_2 und w_3 kann die Verteilung den Bedürfnissen entsprechend korrigiert und beeinflusst werden.

Z Transformation

Bringt geeignete Quelldaten annähernd auf eine Normalverteilung, gemäß der Formel:

$$x_{Trans} = \frac{1}{2} \cdot ln\left(\frac{1+x}{1-x}\right)$$

wobei $x \in (-1, 1)$ sein muss.

Beispiel

Für die Liste $(1, 3, 9, 10, 24, 29, 31, 37, 52, 54, 55)$ wird linear und logarithmisch transformiert (Abbildung 4.62). Bei der linearen Transformation sollen die Werte zwischen 0 und 4 liegen.

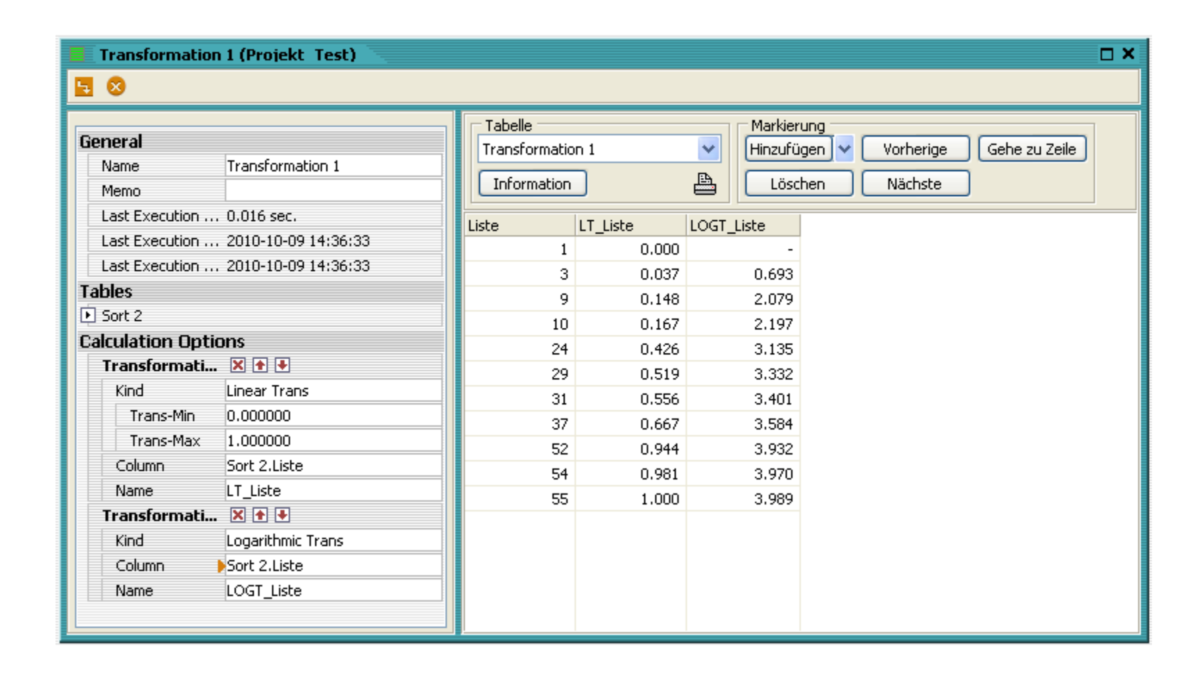


Abbildung 4.62.: Beispiel: linear Transformationlogarithmic Transformation

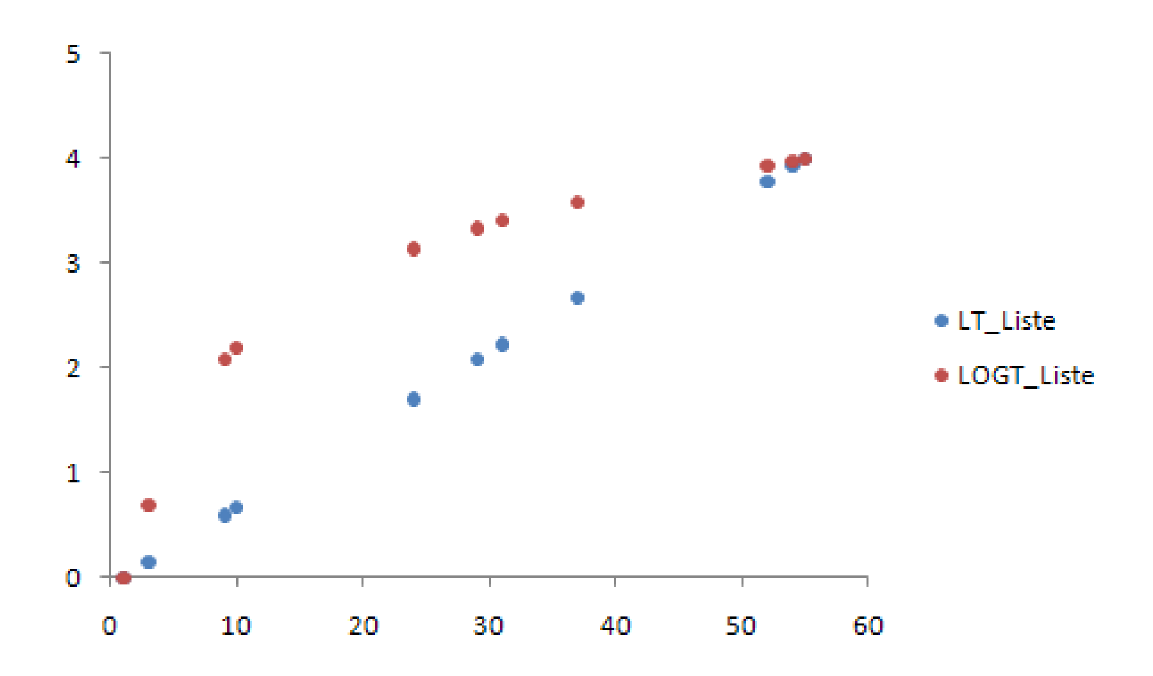


Abbildung 4.63.: Unterschiedliches Verhalten einer linearen (LT) und logaritmischen (LOGT) Transformation

4.3.22. Windowed Aggregation

Kurzbeschreibung	
Verwendungszweck	Aggregation auf dynamisch bestimmte Teilbereiche
Eingang	Eine Tabelle, je eine numerische Spalte pro Funktionsblock für die Aggregation, beliebige zur Parametrisierung
Ausgabe	Eingangstabelle zzgl. je einer Ergebnisspalte
Besonderheit	-

Modul zur Erzeugung von Aggregationen auf dynamisch bestimmten Teilbereichen (sog. Fenster) einer Tabelle. Zu jeder Zeile wird pro erzeugter Spalte genau ein aggregierter Wert berechnet, der sich dynamisch aus den Werten der aktuellen Zeile ergibt.

Bezeichnung	Werte
Aggregation Column	Spalte, über der die Aggregation gebildet werden soll.
Aggregation Kind	Legt die Art der Aggregation fest (Count, Count Valid, Sum, Average, Min, Max, Variance, Standard Deviation, Quad Sum, Quad Average)
Partitions	Legt eine oder mehrere Spalten fest, nach denen die so genannten Partitionen festgelegt werden. Bezogen auf die aktuelle Zeile wird dabei die aktuelle Partition als alle Zeilen in der Quelltabelle definiert, die in ihren Werten in den unterhalb von Partitions eingestellten Spalten übereinstimmen.

Window Mode

Legt den Modus fest, nach dem Fenster innerhalb der Partitionen gebildet werden. Innerhalb dieser Fenster werden dann gemäß der anderen Einstellungen für die aktuellen Zeilen Aggregationen durchgeführt.
Window by Value: Hier werden die Fenster durch ein dynamisch bestimmtes Werteintervall innerhalb der aktuellen Partition definiert. Grundlage für dieses Intervall ist die Auswahl der *Window Column*. Wird für eine Zeile der dazugehörige Aggregationswert bestimmt, so ist der Wert der *Window Column* in dieser Zeile Basis für das Interval.
Die Untergrenze des Intervalls berechnet sich durch
Window Column — Preceding
Die Obergrenze des Intervalls berechnet sich durch
Window Column + Following.
Window by Rownum: Hier werden die Fenster durch einen über die Position der aktuellen Zeile innerhalb der aktuellen Partition definierten festen Bereich der Partition gebildet. Da diese Fenster von der Reihenfolge der Daten in der Quelltabelle abhängen, kann diese zusätzlich mittels der Einstellungen unter Sort manipuliert werden. Die Untergrenze (als Zeilennummer innerhalb der aktuellen Partition zu verstehen) des Intervalls berechnet sich durch
[Aktuelle Zeile] — Preceding.
Die Obergrenze des Intervals berechnet sich durch
[Aktuelle Zeile] + Following.

Preceding

Bestimmt den Bereich, durch den die Untergrenze der dynamischen Teilbereiche bestimmt werden (s.o). Geben Sie direkt die relative Position oder mit den Konstanten *CURRENT_ROW* oder *FIRST_ROW* die aktuelle Position oder die der ersten Zeile der Partition an.

Following	Bestimmt den Bereich, durch den die Obergrenze der dynamischen Teilbereiche bestimmt werden (s.o). Geben Sie direkt die relative Position oder mit den Konstanten *CURRENT_ROW* oder *LAST_ROW* die aktuelle Position oder die der letzten Zeile der Partition an.
Name	Name der zu erzeugenden Spalte

Tabelle 4.22.: Parameter Windowed Aggregation

4.4. Neural Nets

4.4.1. Kohonen

Kurzbeschreibung	
Verwendungszweck	Clusteralgorithmus auf Basis neuronaler Netze zum Auffinden ähnlicher Datensätze
Eingang	Eine Tabelle, je eine Spalte pro Funktionsblock
Ausgabe	Zwei Tabellen: 1. Eingangstabelle zzgl. zweier Spalten mit den Koordinaten der Datensätze auf der Netzkarte 2. Tabelle mit einer zweidimensionalen Spalte mit der Netzkarte und je einer zweidimensionalen Ergebnisspalte pro Block
Besonderheit	-

Das neuronale Kohonennetz besteht aus einer wählbaren Anzahl von Neuronen, die durch die Eingaben *Columns* und *Rows* festgelegt werden: Die Anzahl der Neuronen im Netz errechnet sich, indem man diese beiden Variablen multipliziert.

Jedes Neuron definiert sich durch eine Reihe von numerischen Werten, die Spaltenwerte repräsentieren. Diese Spaltenwerte, durch die sich ein Neuron definiert, sind diejenigen, die man als *Learn and BM Calculation* ausgewählt hat.

Hinweis!
Streng genommen definiert sich jedes Neuron darüber hinaus auch über die Spalten, die man als „Learn only“ ausgewählt hat. Diese haben jedoch keinen Einfluss auf das Erscheinungsbild des Berg- und Tal–Graphik–Ergebnisses, werden nur latent in einem nicht zur Berg–und–Tal–Darstellung genutzten Teil der Ergebnismatrix (UMX) gespeichert. Durch diese Speicherung sind die Variablenausprägungen der Learn–Only–Variablen auf den einzelnen Neuronen später farblich darstellbar.

Da zu Beginn einer jeden Netzanalyse diese Werte zufällig ausgewählt werden, stellt jedes Neuron zu Beginn, wie auch im späteren Verlauf während des Anlernens des Netzes, eine ganz bestimmte „Werte–Realisierung“ dieser Spaltenwerte dar.

Das Anlernen des Netzes besteht darin, jeden einzelnen Datensatz bzw. jede Zeile nacheinander daraufhin zu prüfen, welchem Neuron es in Bezug auf seine Spalteninhalte am ähnlichsten ist und ihn auf diesem Neuron zu platzieren.

Hinweis!
Das Ähnlichkeits- bzw. Distanzmaß ist die sog. euklidische Distanz.

Gleichzeitig werden die Werte dieses Neurons diesem Datensatz „recht stark“ angepasst. Alle umliegenden Neuronen werden diesem Datensatz ebenfalls angeglichen, aber je nach Entfernung in immer geringerem Maße.

Sind alle Datensätze auf die Neuronen verteilt worden, und haben sich die Neuronen diesen Datensätzen in der beschriebenen Weise angeglichen, ist eine „Lernepoche“ abgeschlossen und eine neue kann beginnen.

Die Anzahl an Lernepochen, die das Netz durchlaufen soll, kann vor Beginn der Analyse gewählt werden. Eine Faustregel, wie viele Lernepochen sinnvoll sind, um ein gut interpretierbares Ergebnis zu erhalten, gibt es leider nicht, da es von den Daten direkt abhängt, nach wie vielen Lernepochen ein Netz überlernt ist, d.h. Gruppen künstlich auseinander gezogen und nicht mehr gefunden werden.

Die folgenden Lernepochen verlaufen genauso wie die erste, doch wird das Maß der Stärke der Angleichung der Neurone an die Datensätze bei jeder Lernepoche etwas geringer.

Hinweis!
Die Angleichungsstärke verringert sich linear durch den Faktor $\frac{(N-n+1)}{N}$, wobei N = Anzahl der Lernepochen und n = Rang der betreffenden Lernepoche in der Rangreihenfolge aller Lernepochen.

Ergebnis sichtbar machen

Das Ergebnis des angelernten Kohonen–Netzwerks kann man in der Graphic View sichtbar machen (siehe Abbildung 4.74 Surface auf Seite 201).

Die Ergebnismatrix (UMX) kann im 3D–Grafikmodus mit gedrück-

ter Maustaste auf jede der drei Koordinatenachsen gezogen werden und es erscheint eine Berg- und Tallandschaft auf dem Neuronennetz.

Die Spalten, die zum Anlernen *(Learn only und Learn and BM Calculation)* benutzt worden sind *(WTS)*, können auf die Landschaft gezogen werden und diese je nach Ausprägungsgrad dieser Spalte in einer bestimmten Region anhand der Neuronen–Spaltenwerte einfärben.

Wählt man in den Funktionsblöcken *Columns* und Rows die gleichnamigen Spalten aus, die vom Kohonen–Modul zur Verfügung gestellt werden, erscheinen die einzelnen Datensätze als Klötzchen auf den Kreuzungspunkten bzw. Neuronen des Netzes, wobei große Klötzchen mehr Datensätze repräsentieren als kleine.

Jede numerische Spaltenausprägung der Datensätze/Klötzchen kann nun durch Einfärbung der Klötzchen kenntlich gemacht werden: Hier muss ebenfalls nur die entsprechende Spalte im Funktionsblock Cube *Color* ausgewählt werden. Man beachte, dass man hierbei die Werte der tatsächlichen Datensätze und nicht die Werte der darunter liegenden Neuronen betrachtet (obwohl diese natürlich recht ähnlich sind).

Ergebnisinterpretation

Jeder Würfel, der auf der Netzkarte eingezeichnet ist, repräsentiert eine bestimmte Anzahl an Datensätzen.

Die Ähnlichkeit der Datensätze bzw. der darunter liegenden Neuronen zueinander können anhand der Entfernung voneinander bestimmt werden.

Dabei sind folgende Punkte zu beachten:

- Die Kohonen–Netzkarte hat die Form eines Torus, also wie der Schlauch eines Autoreifens, der für die bessere Darstellung aufgeschnitten und flach ausgebreitet dargestellt wird.

Das bedeutet für die Interpretation, dass die Ergebniskarte theoretisch an den gegenüberliegenden Kanten verbunden ist: Der linke Kartenrand ist mit dem rechten verbunden, so wie der obere in den unteren Kartenrand übergeht.

- Es darf zudem nicht nur die Entfernung auf der zweidimensionalen X–Z–Ebene berücksichtigt werden — so wie es Bergsteiger ja auch nicht tun. Man sollte sich die Berge und Täler wie Falten auf einer Tischdecke vorstellen. Zwei Flecken, die rechts und links von einer Auffältung scheinbar nahe beieinander liegen, sind plötzlich sehr viel weiter voneinander entfernt, wenn die Tischdecke geradegezogen wird. So sind also Würfelgruppen, die in Tälern lokalisiert sind, einander viel ähnlicher gegenüber den Gruppen außerhalb des Tals oder jenseits eines Bergrückens.
  Liegen Datensätze (Klötzchen) nicht an einem Berg, sondern auf dessen Spitze oder recht weit oben am Hang, gilt für diese die Tischdekken–Metapher nicht mehr so ganz: Je weiter oben ein Datensatz auf einem Berg platziert ist, desto alleinstehender bzw. unähnlich ist er gegenüber *allen anderen* Nachbarn.
  Das bedeutet, zu einem am Fuße des Tals liegenden Datensatz ist er dem Tischdeckengleichnis entsprechend weit entfernt, doch ist er womöglich auch weit von Datensätzen entfernt, die mit ihm weit oben am Berg liegen. Ab einer gewissen Berghöhe sind die Abstände der auf dem Berg angesiedelten Fälle in ihrer Ähnlichkeit also nicht mehr rein geometrisch interpretierbar. Ihnen gemeinsam ist vor allem Ihre Unähnlichkeit allen anderen Nachbarn gegenüber.
  Gibt es viele dieser Ausreißer–Datensätze kann es nützlich sein, nur mit diesen eine gesonderte Kohonen–Analyse durchzuführen. Insbesondere, wenn um diesen Berg herum nur flaches Gelände zu finden ist, sich also überhaupt kaum Gruppen in Tälern bilden.

- Bisweilen finden sich spitze Berge mit wenigen verstreuten Datensätzen darauf, die man gern intuitiv als ähnlich anse-

hen möchte, da sie ja alle irgendwo an den Wänden dieses Berges „hängen". Auch hier muss man sich das Auseinanderziehen einer Falte vor Augen halten, um erkennen zu können, dass diese Fälle nur dann auch sehr ähnlich sind, wenn sie in ähnlicher Höhe auf diesem Berg beieinander liegen.

4.4.2. Bedienungshinweise

Die Spalten, die man als *Learn and BM Calculation* auswählt, sind diejenigen Spalten, nach denen die einzelnen Datensätze gruppiert werden sollen.

Die Spalten, die man als *Learn and BM Calculation* und *Learn only* ausgewählt hat, können auf der „Ergebnislandschaft" farblich aufgetragen werden.

Alle restlichen numerischen Spalten können in Form von Würfeleinfärbungen in der Ergebnislandschaft dargestellt werden und damit die Spaltenwerte der Datensätze selbst kenntlich machen.

Eine Empfehlung für die optimale Anzahl Lernepochen zu geben, ist nicht ohne weiteres möglich. Zu wenige Lernepochen trennen tatsächlich vorhandene Gruppen nicht gut genug voneinander, zu viele Lernepochen führen zu einem „Überlernen" des Netzes, was tatsächlich existierende Gruppen wieder auseinanderdifferenzieren würde.

Folgende Hinweise sollen nur grobe Anhaltspunkte für die Wahl der Lernepochen–Anzahl liefern:

- Je mehr Datensätze man zum Anlernen verwendet, desto weniger Lernepochen sind nötig.
- Je deutlicher die Abgrenzungen der Gruppen erwartet werden oder entdeckt werden sollen, desto weniger Lernepochen sind nötig.
- Zusammenhänge, die bei sehr wenigen Lernepochen (1 bis 3) gefunden werden, sich bald darauf jedoch wieder auflö-

sen, könnten Zufallsergebnisse sein. Sie sollten vor einer Interpretation in jedem Fall mehrfach repliziert werden — wo keine Gruppen sind, sollte man auch keine sehen wollen.

- Gruppen, die auch nach vielen Lernepochen noch sichtbar sind, sind empirisch existent.
- Hat man nicht allzu viele Datensätze, sind Ergebnisse bei 10 bis 15 Lernepochen meist gut replizierbar.

Das Kohonen–Modul erzeugt zwei Tabellen:

1. Quelldaten, erweitert um die Best–Match–Positionen auf der Karte.
2. Die vom Kohonen berechneten Matritzen.

Beispiel

Betrachten wir eine dreidimensionale Datenmenge, dessen Verteilung in Abbildung 4.64 sichtbar ist.

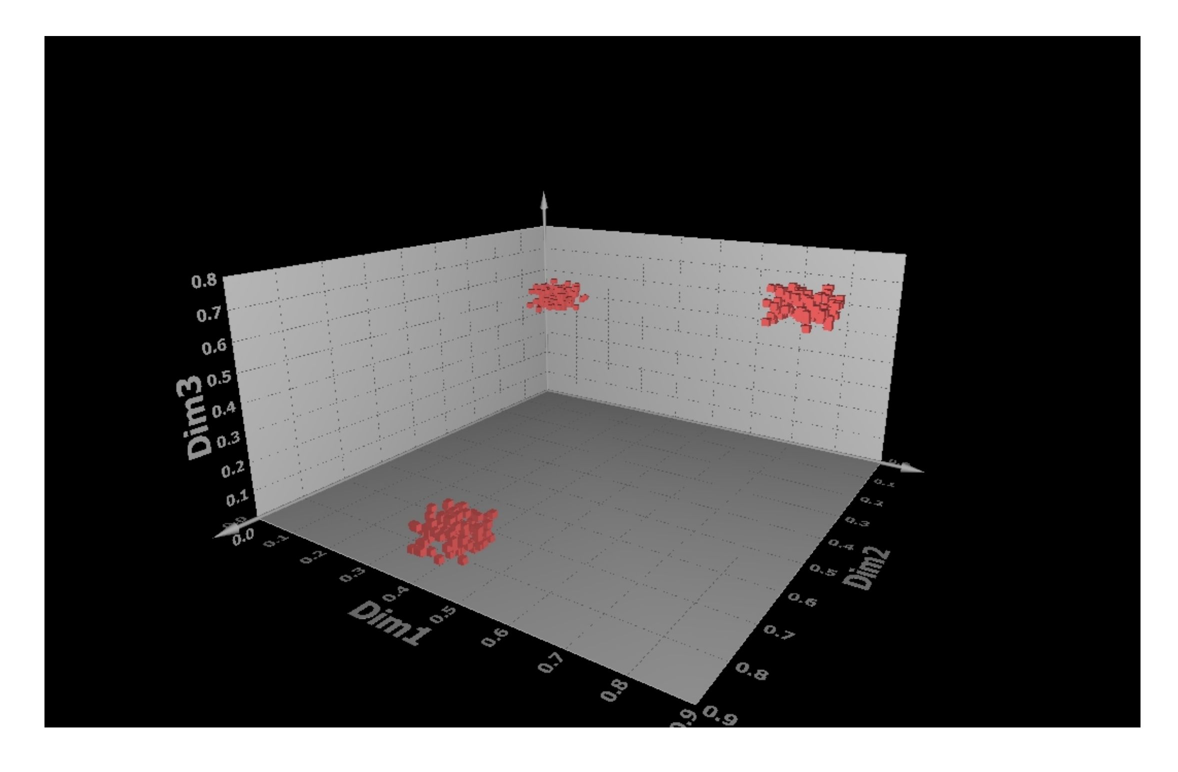


Abbildung 4.64.: Verteilung der Datenmenge

Es handelt sich um drei Datenhaufen, auf die wir das Kohonen–Modul anwenden.

Das Kohonen–Modul wir wie in Abbildung 4.65 eingestellt.

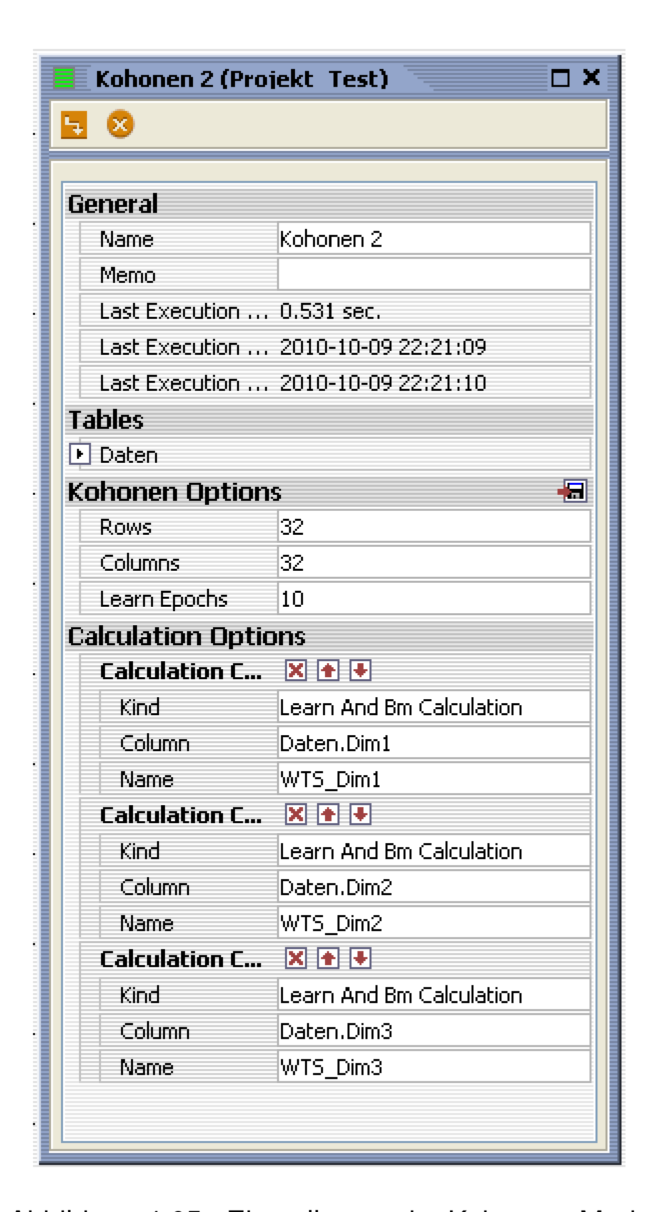


Abbildung 4.65.: Einstellungen im Kohonen–Modul

Logischerweise entstehen nun drei Täler (Abbildungen 4.66 und 4.67). Jedes der Täler repräsentiert einen Datenhaufen aus der Abbildung 4.64.

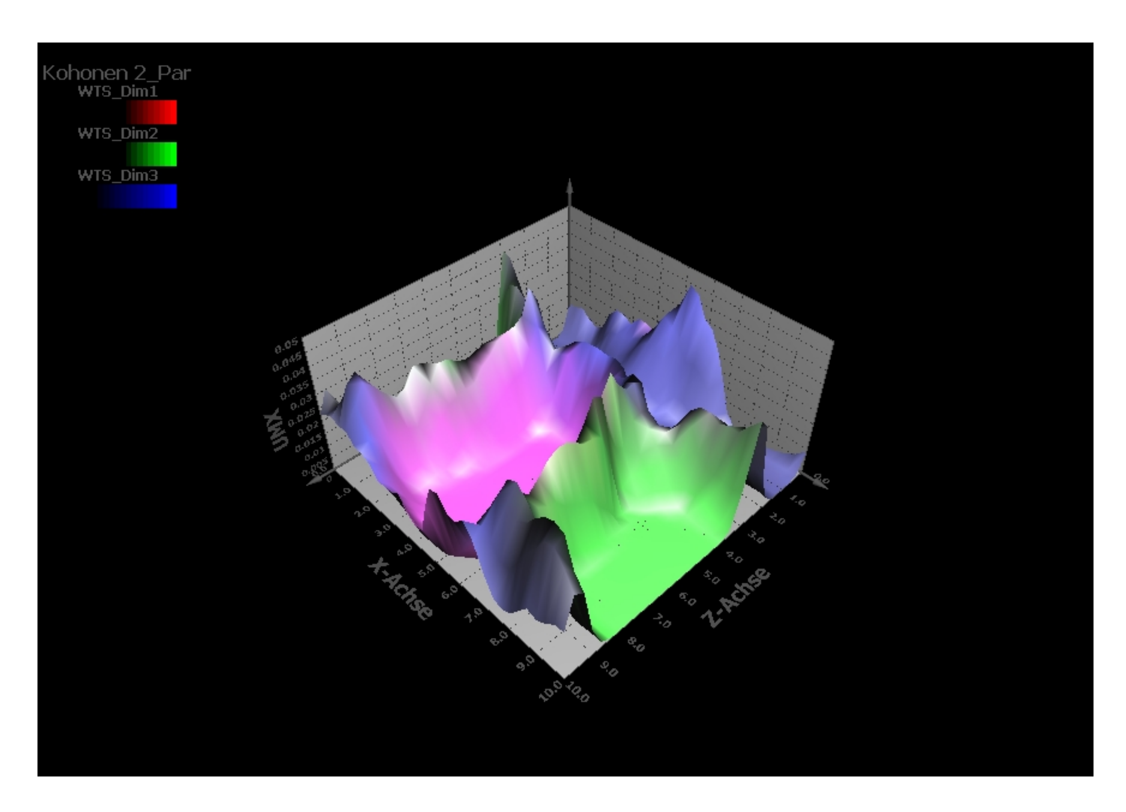


Abbildung 4.66.: Graphisches Ergebnis des Moduls Kohonen

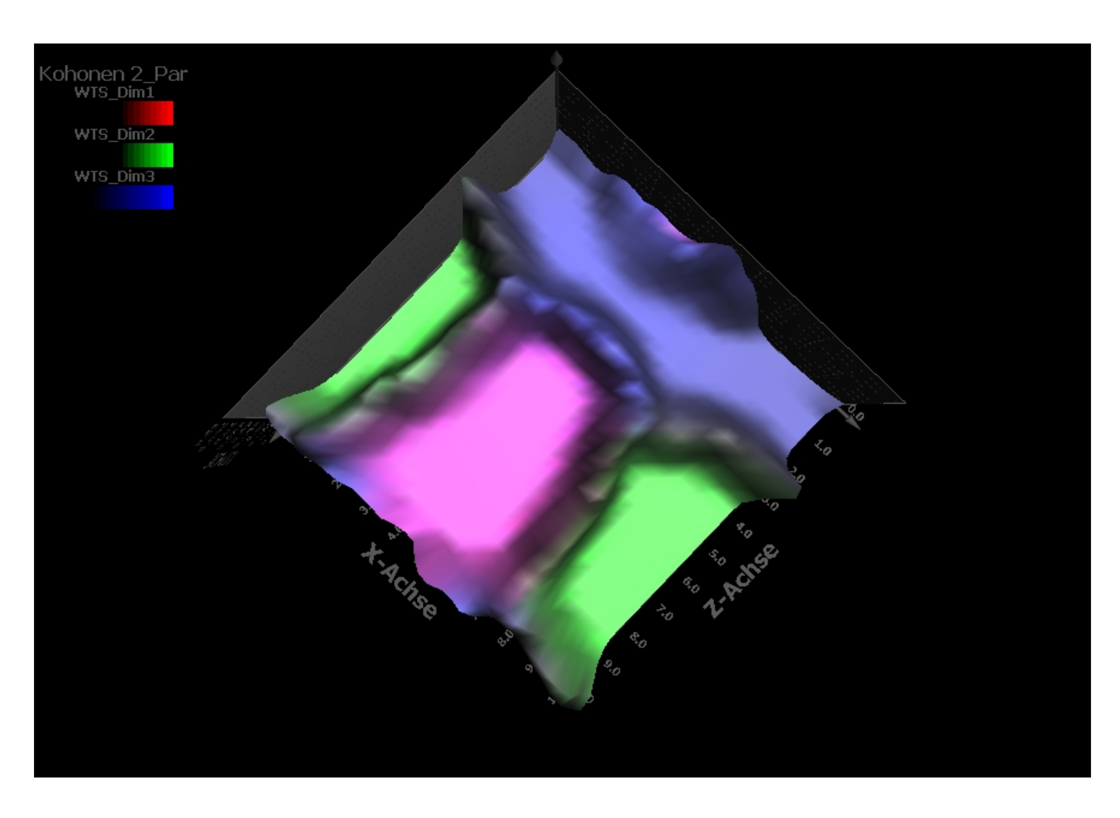


Abbildung 4.67.: Graphisches Ergebnis des Moduls Kohonen (Ansicht von oben)

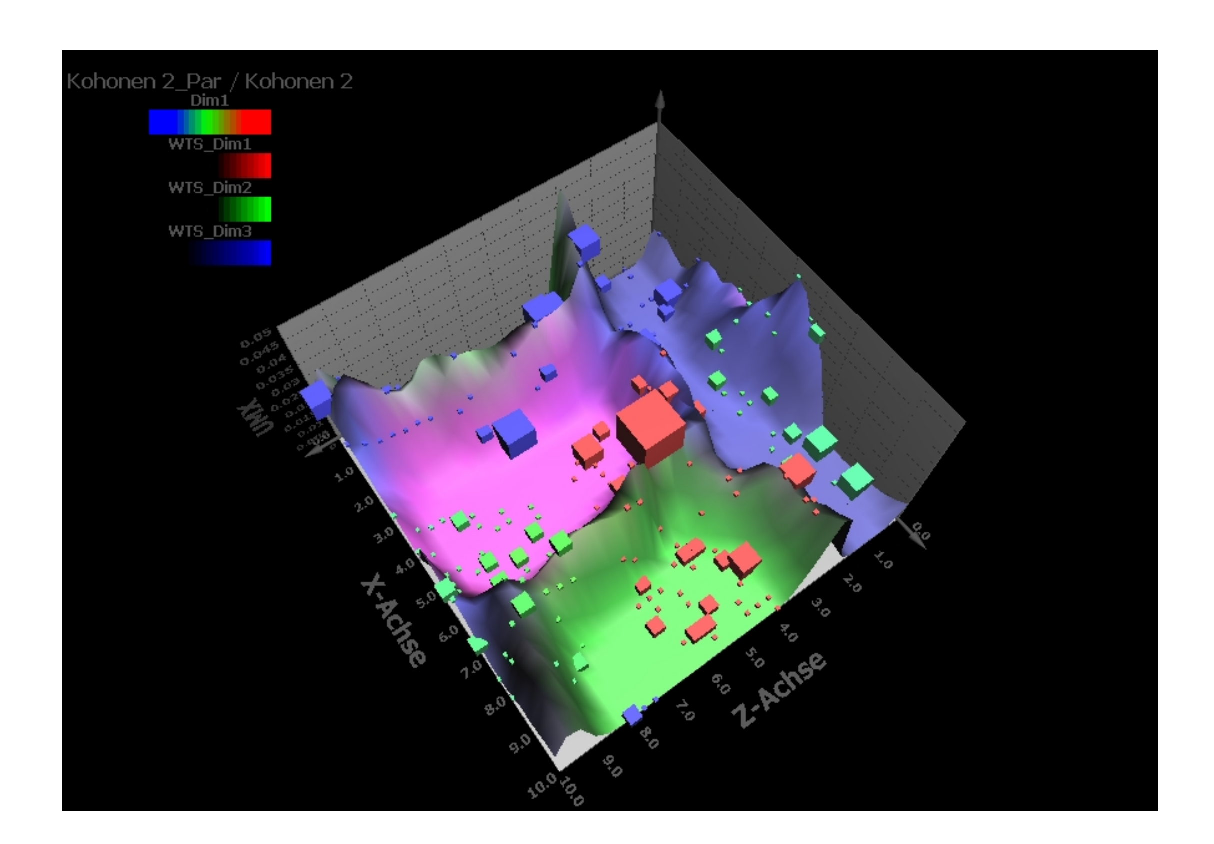


Abbildung 4.68.: Graphisches Ergebnis mit Klötzchen

4.5. Trigger

Trigger stellen eine Sonderform innerhalb der Modul–Palette dar, da sie nicht mit Tabellen arbeiten, sondern lediglich eine steuernde Funktion auf andere Module ausüben. Die Links zwischen diesen und anderen Modulen symbolisieren in diesem Fall nur eine Steuerung und keinen Datenfluss.

4.5.1. Timer

Ein Timer führt ein mit ihm verbundenes Modul zu einstellbaren Zeiten aus. Dies geschieht auch dann, wenn der Designer nicht aktiv ist und das Projekt vom Server eigenständig ausgeführt wird.

Kurzbeschreibung	
Verwendungszweck	Zeitgesteurtes Starten von Analysen
Eingang	-
Ausgabe	-
Besonderheit	Die Verbindung zu anderen Modulen startet den Analysevorgang.

Ein Timer führt ein mit ihm verbundenes Modul zu einstellbaren Zeiten aus. Dies geschieht auch dann, wenn der Designer nicht aktiv ist und das Projekt vom Server eigenständig ausgeführt wird.

Timer–Ereignis verwalten

Klicken Sie auf die Schaltfläche *Timer hinzufügen*, um einen neuen Trigger hinzuzufügen. Nehmen Sie die gewünschten Änderungen vor und speichern diese mit *Übernehmen* ab.

Mit Timer *entfernen* löschen Sie ein Ereignis wieder.

Timer–Einstellungen

Es stehen grundsätzlich zwei verschiedene Modi zur Verfügung:

1. Modus *Einmalig ausführen* führt ein verbundenes Modul einmalig zu den jeweils angegebenen Zeiten aus. In diesem Modus können Sie zum Beispiel einen nächtlichen, rechenintensiven Prozess starten.

2. *Modus Wiederholen* erlaubt ein wiederholtes Ausführen der verbundenen Module in einer Schleife mit der angegebenen Loop–Zeit *(Wiederkehrend)* zu den jeweils abgegebenen Zeiten. Ist keine Endzeit angegeben, wird die Schleife endlos ausgeführt, bis sie vom Benutzer abgebrochen wird.

Innerhalb der beiden beschriebenen Modi sind nahezu alle Ausführungskombinationen möglich.

Datum und Zeit beschränken. *Datum aktivieren* erlaubt die Vorwahl eines bestimmten Datums, an oder ab dem der Timer aktiviert sein soll, mit *Uhrzeit aktivieren* kann zusätzlich eine Uhrzeit gewählt werden. Im Modus Run Once wird an diesem Datum das verbundene Modul genau einmal zur angegeben Startzeit ausgeführt. Bei *Wiederholen* wird das verbundene Modul zwischen dem angegeben Start– und Enddatum jeweils zwischen der gewählten Start– und Endzeit im festgelegten Loop–Intervall ausgeführt.

Statt eines festen Datums können Sie alternativ mit der Option Recurrent einen Wochen- und Monathsrhythmus festlegen. Mit *Wochentage aktivieren* begrenzen Sie die Ausführung auf bestimmte Wochentage, mit *Kalender aktivieren* auf bestimmte Tage innerhalb eines Monats.

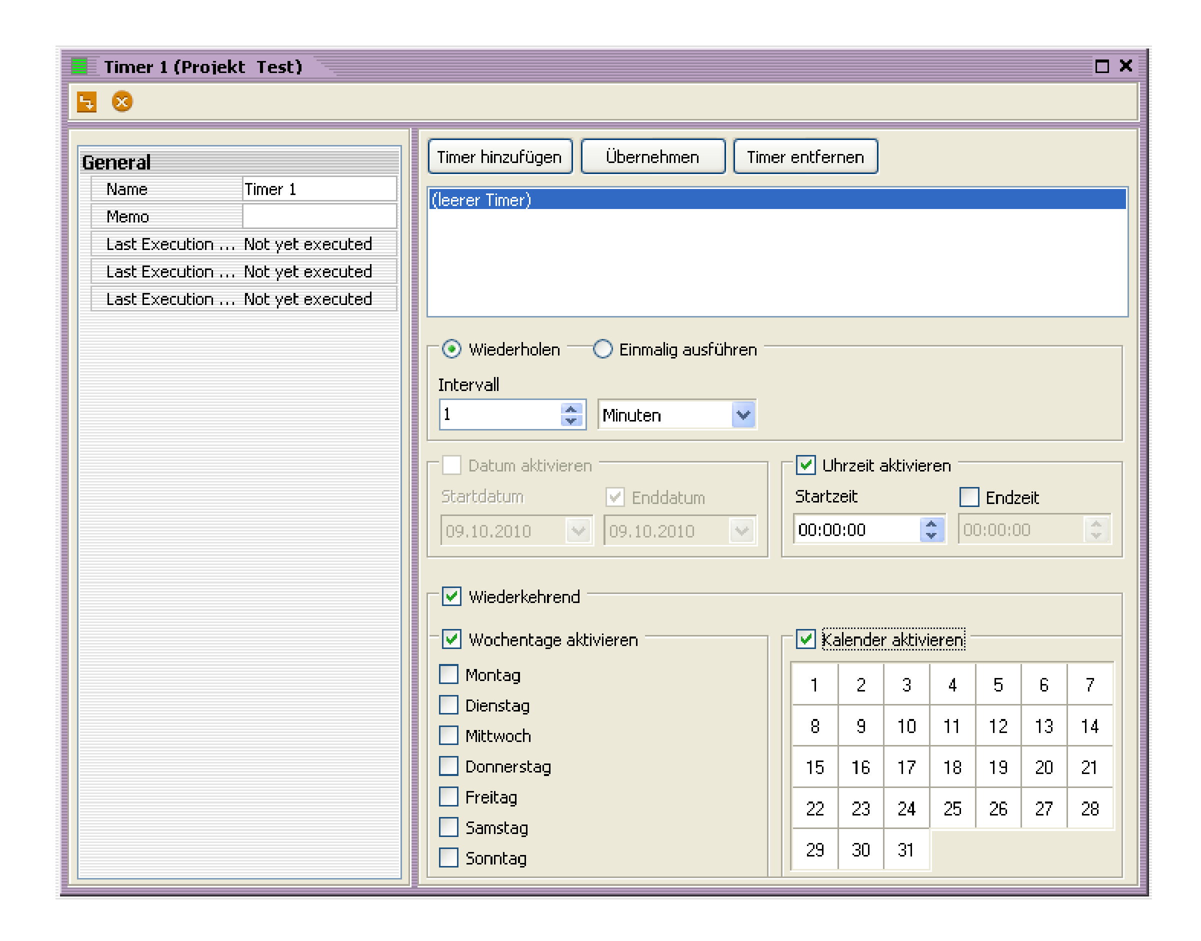


Abbildung 4.69.: Timer: Einstellungen

Hinweis!
Bei der Ausführung des Timer–Ereignisses werden alle Execute–Prozeduren des verbundenen Moduls und aller davorliegenden gestartet, unabhängig davon, ob bereits gültige Daten vorliegen. Damit wird sichergestellt, dass bei jedem Durchlauf mit aktualisierten Daten gearbeitet wird.

4.5.2. Database Trigger

Kurzbeschreibung	
Verwendungszweck	Starten von Analysen auf Datenbankereignisse
Eingang	-
Ausgabe	-
Besonderheit	Die Verbindung zu anderen Modulen startet den Analysevorgang.

Das Database Trigger Modul führt ein mit ihm verbundenes Modul aufgrund von Ereignissen in einer selektierten Datenbank aus. Dies geschieht auch dann, wenn der Designer nicht aktiv ist und das Projekt vom Server eigenständig ausgeführt wird.

Wichtig!
Der Database Trigger setzt eine Oracle–Datenbank ab Version 8 voraus!

Basis des Database Triggers ist ein auf dem Oracle–Server eingerichtetes Advanced Queuing (AQ). Informationen über die Ein-

richtung und Verwaltung des Advanced Queuings entnehmen Sie bitte dem „Oracle Application Developer's Guide — Advanced Queuing“ der Firma Oracle.

Einrichtung

Oracle stellt verschiedene Queue–Typen zur Verfügung. Hier werden nur die notwenigen Schritte zum Einrichten anhand einer von Oracle als „nicht persistent“ bezeichneten Queue beschrieben. Aber auch Messages anderer Queue–Typen können verarbeitet werden („Normal Queue“, „Exception Queue“).

Zunächst wird eine in RayQ eingetragene Datenbank unter *Database* ausgewählt. Alle vom Administrator in RayQ mit einem nativen Oracle Treiber registrierten Datenquellen werden in der Auswahlbox angezeigt.

Nach Auswahl einer Oracle Datenquelle erscheinen die *Location* und *Driver Description* der Datenquelle. Geben Sie unter *Authentication* die Anmeldedaten ein, werden unter *Available Queues* alle zur Verfügung stehenden Queues angezeigt. Nachdem Sie eine dieser Queues ausgewählt haben, müssen Sie noch den in der Oracle–Datenbank eingerichteten Subscriber im Feld *Subscriber* eintragen.

Durch das Starten des Moduls mit *Execute* wird der Trigger aktiviert und befindet sich in Bereitschaft.

Bezeichnung	Werte
Database	„Net Service Name“ der Oracle–Datenbank oder des Oracle–Dienstes
Driver Description	Treibername
Authentication	Benutzer der Oracle–Datenbank

Available Queues	Liste der verfügbaren Queues deren Messages verarbeitet werden sollen. Eine entsprechende Queue wird z.B. mit dem Befehl „DBMS_AQADM.CREATE_NP_QUEUE“ erzeugt. Nachdem die Queue mit dem Befehl „DBMS_AQADM.START_QUEUE“ aktiviert wurde, wird sie in der Liste aufgeführt. Ohne weitere Schritte auf dem Datenbank–Server können allerdings weiterhin keine Messages empfangen werden.
Name der Queue	Setzen Sie einen Haken bei den Queues, dessen Nachrichten von RayQ empfangen werden sollen
Subscriber	Name des „Subscribers“, der mit dem Befehl „DBMS_AQADM.ADD_SUBSCRIBER“ zur Queue hinzugefügt wurde.

Tabelle 4.23.: Parameter Database Trigger

Funktionstest

Starten Sie das Modul mit dem Execute–Button. Das Modul wartet nun auf eine neue Nachricht der Oracle–Datenbank. Setzen Sie auf der Datenbank mit dem Befehl „DBMS_AQ.ENQUEUE“ eine solche Nachricht ab. Sobald das Database Trigger Modul die Nachricht empfängt, startet es die Ausführung aller verbundenen Module.

Mit *Stop* beenden Sie die Bereitschaft des Triggers.

Wichtig!
Vor der Verwendung eines Oracle Database Triggers muss die jeweilige Datenquelle im RayQ Server registriert sein (siehe 5.2.4 Databases auf Seite 222).

Tipp!
Ein vollständiges Beispiel–SQL–Skript zum Erzeugen einer Message–Queue befindet sich im Verzeichnis „AddOns“ auf dem Datenträger der RayQ–Server–Version.

Hinweis!
Die Einrichtung von Advance Queuing sollte nur von erfahrenen Oracle–Datenbankadministratoren erfolgen.

4.5.3. Socket Listener

Kurzbeschreibung	
Verwendungszweck	Starten von Analysen auf Netzwerknachrichten
Eingang	-
Ausgabe	-
Besonderheit	Die Verbindung zu anderen Modulen startet den Analysevorgang.

Das Socket Listener Modul führt ein mit ihm verbundenes Modul aufgrund von Ereignissen im Netzwerk an einem einstellbaren Listener Port aus. Dies geschieht auch dann, wenn der Designer nicht aktiv ist und das Projekt vom Server eigenständig ausgeführt wird.

Bezeichnung	**Werte**
Port	Port–Nummer, auf die das Modul Nachrichten empfangen soll
Message	Netzwerk–Nachricht, auf die das Modul reagieren soll

Tabelle 4.24.: Aggregat–Funktionen Pivot

Trifft an dem spezifizierten Port die eingestellte Message ein, so startet der oben beschriebene Prozess.

Insbesondere kann das Modul auf Ereignisse in einer Microsoft SQL–Server Datenbank reagieren, wenn dort mittels einer Extended Stored Procedure innerhalb eines Triggers Messages an den oben spezifizieten Port gesand werden.

4.6. Export

Export–Module dienen dazu, die in RayQ berechneten Werte nach außen zur Verfügung zu stellen oder abzuspeichern.

4.6.1. File Export

Kurzbeschreibung	
Verwendungszweck	Lokales Exportieren von Tabellen
Eingang	Eine Tabelle, beliebig viele Spalten beliebigen Typs
Ausgabe	-
Besonderheit	Dateien werden lokal gespeichert, die Ausführung ist nur so lange möglich, wie der RayQ Designer geöffnet ist.

Das Export–Modul erlaubt es, Tabellen in Dateien auf einen lokalen Datenträger zu exportieren. Ihnen stehen dafür die vier gängigsten Ausgabeformate zur Verfügung: Text (CSV), Excel, HTML und XML.

Setzen Sie unter *Tables* an einer oder mehrerer Spalten einen Haken, um sie für den Export auszuwählen. Wählen Sie im Parameter *File* einen Dateinamen für die Ausgabe. Abhängig von der Endung wird das entsprechende Format generiert.

Hinweis!
Der File–Export arbeitet lokal auf Ihrem Client, kann demnach nur so lange ausgeführt werden, wie der RayQ–Designer aktiv ist. Möchten Sie einen Export ausführen, der unabhängig vom Client auch nach dem Beenden des Designers arbeitet, so verwenden Sie das Modul „Server Export“ (siehe 4.6.3, Seite 189).

4.6.2. RayQ Data Export

Kurzbeschreibung	
Verwendungszweck	Speichern eines Abbildes einer Tabelle
Eingang	Eine Tabelle, beliebig viele Spalten beliebigen Typs
Ausgabe	-
Besonderheit	Nach dem Speichern des Abbildes steht eine neue Datenquelle zur Verfügung, die diese Daten enthält.

Das Modul erzeugt eine neue Datenquelle, die als eigenständiges Modul unter „Local Data“ in der Modul–Leiste zur Verfügung steht.

Wählen Sie unter *Tables* die zu exportierenden Spalten einer Tabelle aus. Vergeben Sie unter *Table Name* einen Namen des neu zu erzeugenden Moduls.

Mit dem Ausführen des Moduls wird eine RQD–Datendatei auf dem Server erzeugt, die die exportierte Tabelle enthält.

4.6.3. Server Export

Kurzbeschreibung	
Verwendungszweck	Exportieren von Tabellen über den Server
Eingang	Eine Tabelle, beliebig viele Spalten beliebigen Typs
Ausgabe	-
Besonderheit	Dateien können unabhängig vom RayQ Designer in die vom Administrator freigegebenen Verzeichnisse gespeichert werden.

Der Server–Export arbeitet vergleichbar mit dem „File Export“ — die Daten werden jedoch unabhängig vom RayQ Designer durch den Server generiert.

Wählen Sie unter Tables die zum Export bestimmten Spalten aus. Unter *Export Options* sind anschließend weitere Einstellungen notwendig

Wichtig!
Der Server Export kann nur dann durchgeführt werden, wenn der Server–Administrator Verzeichnispfade für die Speicherung der Dateien freigegeben hat.

Tipp!
Setzen Sie die Option „Override Existing“, wenn Sie einen automatischen Prozess starten möchten, der in bestimmten Intervallen das aktuelle Analyseergebnis in derselben Datei zur Verfügung stellen soll.

Bezeichnung	Werte
Export Path	Verzeichnis–Pfad, in dem die Datei geschrieben werden soll. *Die zur Verfügung stehenden Pfade müssen vom Server–Administrator festgelegt werden und können nur mit dem RayQ Administrator verändert werden. Siehe 6.2 Konfiguration auf Seite 234.*
File Name	Dateiname, unter dem die Tabelle gespeichert werden soll.
Override Existing	Eventuell vorhandene Datei ohne Warnung überschreiben.

File Type	Auswahl der Dateiformate: *Excel:* XLS–Datei im Microsoft–Excel–Format *XML:* Extensible Markup Language *HTML:* Export als HTML–Tabelle *CSV:* Textdatei als kommaseparierte Liste
Stylesheet (nur bei HTML)	Dateinamen des zugehörigen Stylesheets. Ist keine Datei mit diesem Namen vorhanden, wird vom Server ein neues Stylesheet mit Standard–Angaben erzeugt.
Override Existing (nur bei HTML)	Eventuell vorhandenes Stylesheet immer durch das vom Server generierte ersetzen.
Separator (nur bei CSV)	Separator–Zeichen, mit dem die einzelnen Werte voneinander abgetrennt werden sollen. Voreingestellt ist das Semikolon.

Tabelle 4.25.: Parameter Server Export

Möchten Sie einen periodischen Export von Analyseergebnissen starten, so verwenden Sie neben dem Server Export noch einen Trigger (siehe 4.5, Seite 180). Starten Sie den Trigger und beenden den RayQ Designer mit der Option Projekt im Hintergrund ausführen (siehe 2.4.1 Projekt–Verarbeitung im Hintergrund auf Seite 29 und 3.7.1 Optionen beim Schließen auf Seite 57).

4.7. Templates

Als Templates stehen Ihnen alle Container–Module zur Verfügung, die auf dem Server abgelegt sind. Ein Container ist praktisch ein eigenes Projekt innerhalb eines Projekts, enthält also eine oder mehrere Datenquellen und beliebige Verarbeitungsmodule, die beliebig miteinander verknüpft und parametrisiert sind.

4.7.1. Container erzeugen

Um einen eigenen Container herzustellen, klicken und ziehen Sie mit der Maus auf dem Workspace einen Markierungsrahmen um

die gewünschten Module und wählen im Menü *Bearbeiten / Container herstellen*. Die markierten Module verschwinden daraufhin vom Workspace und ein Container–Modul tritt an ihre Stelle, das alle eventuell vorhandenen Links nach außen beibehält. Voraussetzung ist stets, dass der Container mindestens eine Datenquelle enthält, keine eingehenden Links besitzt und alle enthaltenen Module korrekt parametrisiert sind. Sollten diese Bedingungen nicht erfüllt sein, erhalten Sie beim Aufruf der oben genannten Funktion eine Fehlermeldung. Es ist ebenfalls nicht möglich, einen Container in einen neuen Container einzubinden.

Hinweis!
Das Container–Modul stellt nur die Tabellen von zuvor ausgewählten Modulen nach außen zur Verfügung. Der im View–Fenster dargestellte Container–Workspace besitzt in der linken oberen Ecke ein Symbol, das den Weg „aus dem Container“ darstellen soll. Mit ihm können Sie alle im Container enthaltene Module in bekannter Weise verlinken.

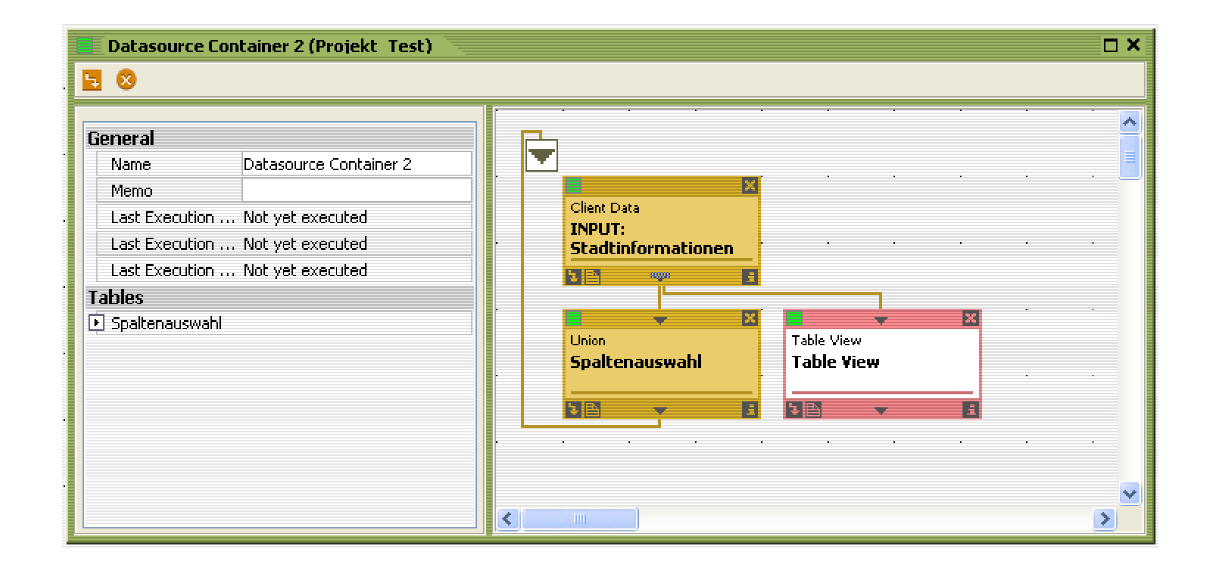


Abbildung 4.70.: View–Fenster eines Containers

Öffnen Sie das View–Fenster eines Containers (Abbildung 4.70), so können Sie in ihn „hineinblicken“. Im eigenen Workspace lassen sich die enthaltenen Module genauso bearbeiten wie auf dem allgemeinen.

4.7.2. Container speichern

Diese Container–Module lassen sich auf dem Server speichern, um sie als eigenständige Module allgemein zur Verfügung zu stellen. Selektieren Sie den gewünschten Container und wählen die Menü–Funktion *Bearbeiten/Container Speichern*.

Der auf diese Weise neu entstandene Modultyp wird in der Modul–Leiste angezeigt und lässt sich wie jedes andere Modul auf den Workspace ziehen. Dort verhält sich das Modul wie der zuvor gespeicherte Container, es kann demnach auch „innen“ verändert werden.

Hinweis!
Als Template gespeicherte Container stehen allen anderen Benutzern zur Verfügung. Auf diese Weise können Sie häufig benötigte Teile oder auf bestimmte Anforderungen angepasste Datenquellen im Netzwerk allen RayQ–Anwendern bereitstellen.

Änderungen in einem auf den Workspace gezogenen Container beziehen sich immer nur auf das betreffende Modul. Die gespeicherte Struktur des als Modultyp in der Modul–Leiste verfügbaren Containers bleibt stets unberührt.

4.8. Views

View–Module bilden die „Schnittstelle“ zum Benutzer, da sie die verarbeiteten Daten visualisieren — grafisch oder tabellarisch. Ein View–Modul kann an beliebiger Stelle in eine Analyse eingefügt werden und hat keinerlei Einfluss auf die eigentlichen Daten. Eingehende Tabellen werden stets unverändert durchgeleitet.

4.8.1. Table View

Kurzbeschreibung	
Verwendungszweck	Tabellenansicht
Eingang	Eine Tabelle mit beliebig vielen eindimensionalen Spalten
Ausgabe	-
Besonderheit	Das Modul ist standardmäßig in jedem Calculation–Modul verfügbar.

Zur Ausgabe von alphanumerischen Daten ist die Table–View (Abbildung 4.71) konzipiert. Table–Views sind als eigenständige Module, sowie im View–Fenster eines Calculation–Moduls verfügbar.

In der Auswahlbox am oberen Rand sehen Sie eine Auflistung aller im Modul verfügbaren Tabellen. Die jeweils ausgewählte wird in der Tabelle dargestellt.

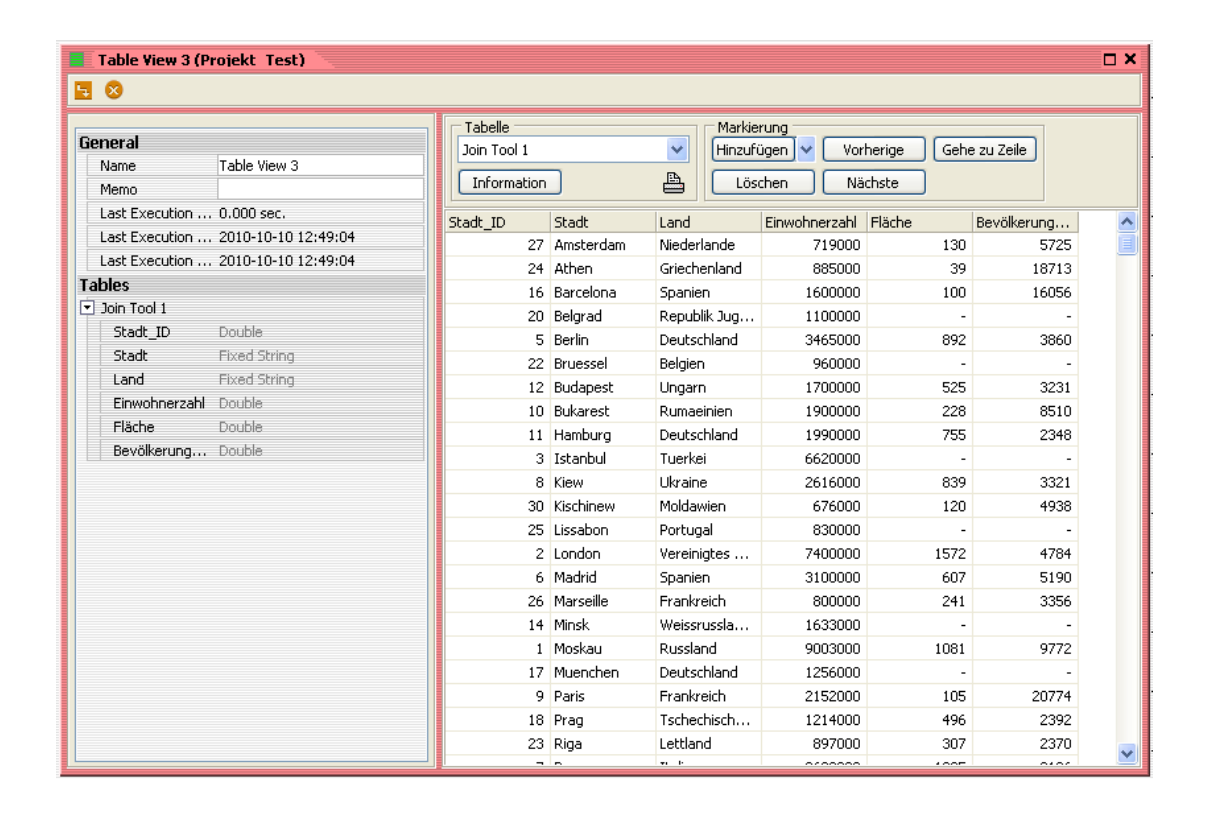


Abbildung 4.71.: Table–View

Tipp!
Die von Calculation–Modulen generierten Tabellen erscheinen in der Auswahlbox häufig erst an zweiter oder dritter Stelle, so dass beim ersten Öffnen einer Calculation View zunächst nur die eingehende Tabelle dargestellt wird. Um das Ergebnis Ihrer Berechnung sehen zu können, ist es daher notwendig, in der Auswahl auf die vom Modul generierte Tabelle umzuschalten.

Um auch größere Datenmengen beherrschen zu können, werden immer nur die Zeilen aus einer Tabelle geladen, die in der Tabelle aktuell angezeigt werden. Aufgrund dieses Verfahrens kann es beim Blättern durch die Tabelle zu kleinen Verzögerungen in der Anzeige kommen.

Hinweis!
Werden keine Daten angezeigt, ist entweder noch keine Analyse ausgeführt worden oder die Ergebnismenge ist z.B. aufgrund eines fehlerhaften SQL–Statements leer.

Zeilen markieren

Einzelne Zeilen einer Tabelle lassen sich markieren. Im Table–View werden diese Zeilen blau hinterlegt dargestellt. Um eine Markierung auf eine oder mehrere Zeilen einzufügen, selektieren Sie sie mit der Maus (ggf. mit gedrückter Umschalt– oder Strg–Taste für Mehrfachselektionen) und wählen die Funktion *Markierung / Hinzufügen*. Zum Ersetzen der vorhandenen Markierung durch die selektierten Zeilen verwenden Sie die Funktion *Markierung / Ersetzen*, zum Entfernen der selektierten entsprechend *Markierung / Entfernen*, die Sie über die Schaltfläche neben dem Funktions–Button umschalten können.

Die Schaltfläche *Löschen* entfernt alle Markierungen innerhalb der Tabelle.

Hinweis!
Eine Änderung der Markierung wird in der betreffenden Tabelle im gesamten System und über mehrere Module hinweg ausgeführt. Das bedeutet, dass alle View–Module die Markierungen dieser Tabelle parallel anzeigen.

Markierte Zeilen finden

Um insbesondere bei großen Tabellen leichter zu einer markierten Zeile gelangen zu können, verwenden Sie die Funktionen *Vorherige* und *Nächste*, die zum jeweils nächsten oder vorherigen markierten Block führt.

Spalten anordnen

Klicken Sie mit der Maus auf einen Spaltentitel und halten die Maustaste gedrückt, lässt sich die Spalte an eine beliebige Position in der Tabelle verschieben. Klicken Sie auf die Trennlinie zwischen zwei Spaltentiteln und halten die Maustaste gedrückt, können Sie die Breite der links angrenzenden Spalte verändern. Die Anordnung und Größen werden im Modul gespeichert und werden auch bei Veränderungen innerhalb der Tabelle wiederhergestellt. Später neu hinzugefügte Spalten werden stets am rechten Ende der Tabelle angefügt.

Hinweis!
Eine Sortierung kann aufgrund der nicht abschätzbaren Datenmenge von der Table–View nicht durchgeführt werden und würde zudem einen Eingriff in die Struktur der Tabelle darstellen. Verwenden Sie hierzu bitte das dafür vorgesehene Sort–Modul (siehe 4.3.17, Seite 156), das für diesen Zweck universell einsetzbar ist.

Tipp!
Drücken Sie die Tastenkombination „Strg“ und „+“, werden alle Spalten auf eine optimale Breite eingestellt.

Tipp!
Um die Übersicht noch weiter zu erhöhen, können Sie nicht benötigte Spalten in der Anzeige ausblenden. Klicken Sie mit der rechten Maustaste auf einen der Spaltentitel. Sie erhalten nun die Möglichkeit, die angewählte Spalte auszublenden *(Spalte verbergen)*, bestimmte Spalten zur Anzeige auszuwählen *(Mehrfache Spaltenauswahl ...)*, alle ausgeblendete Spalten wieder anzeigen *(Alle Spalten anzeigen)* oder die Anzeige auf die ausgwählte Spalte zu reduzieren *(Alle anderen Spalten verbergen)*. Alle diese Einstellungen wirken sich lediglich auf die Anzeige aus, die Spalten und Daten der Tabelle bleiben davon unberührt.

Formatierung ändern

Abhängig vom jeweils verwendeten Datentyp einer Spalte lässt sich die angezeigte Formatierung verändern. Ganzzahlen *(Integer value)* und Zeichenfolgen *(Strings)* besitzen eine feste Formatierung, bei Gleitkommazahlen *(Floating value)* und Datumswerten *(Date/Time)* können Sie aus verschiedenen Formaten wählen. Klicken Sie mit der rechten Maustaste auf den Titel der zu formatierenden Spalte. Im Kontext–Menü wird das jeweils aktive Format angezeigt. Wählen Sie aus dem jeweiligen Untermenü die gewünschte Darstellungsform aus.

Hinweis!
Eine Änderung der Formatierung wirkt sich lediglich auf die Darstellung innerhalb der Tabellenanzeige aus. Die Daten bleiben unverändert in der größtmöglichen Genauigkeit bestehen.

Spalteninformationen anzeigen

Klicken Sie auf die Schaltfläche *Information*, um die Spaltentitel von der einfachen Anzeige des Spaltennamens auf die detaillierte Anzeige umzuschalten. Die detallierte Anzeige (Abbildung 4.72) enthält Informationen über den jeweiligen Spaltentyp, die Anzahl der Zeilen, die Anzahl der gültigen Zeilen, den minimalen und maximalen Wert der gesamten Spalte, den Durchschnittswert, die Varianz und die Summe aller Spaltenwerte.

Stadt_ID	Stadt	Land	Einwohnerzahl	Fläche	Bevölkerungsdichte
Spaltentyp: Double	Spaltentyp: Fixed String[256]	Spaltentyp: Fixed String[256]	Spaltentyp: Double	Spaltentyp: Double	Spaltentyp: Double
Zeilenzahl: 30	Zeilenzahl: 30	Zeilenzahl: 30	Zeilenzahl: 30	Zeilenzahl: 30	Zeilenzahl: 30
Gültige Zeilen: 30	Gültige Zeilen: 30	Gültige Zeilen: 30	Gültige Zeilen: 30	Gültige Zeilen: 22	Gültige Zeilen: 22
Minimum: 1,00			Minimum: 676000,00	Minimum: 0,00	Minimum: 0,00
Maximum: 30,00			Maximum: 9003000,00	Maximum: 1572,00	Maximum: 20774,00
Durchschnitt: 15,50			Durchschnitt: 2200400,00	Durchschnitt: 535,73	Durchschnitt: 6003,32
Varianz: 74,92			Varianz: 4135841240000,00	Varianz: 174662,65	Varianz: 29112492,13
Summe: 465,00			Summe: 66012000,00	Summe: 11786,00	Summe: 132073,00

Abbildung 4.72.: Spalteninformationen im Table View

Tipp!
Klicken Sie mit der rechten Maustaste auf den Spaltentitel, können Sie im Kontext–Menü mit der Funktion *Spalteninformation in Zwischenablage kopieren* die angezeigten Informationen in die Zwischenablage kopieren, um sie an beliebiger Stelle weiterverwenden zu können.

4.8.2. Matrix View

Zur Darstellung der von RayQ verwendeten zweidimensionalen Spalten steht das Modul *Matrix View* zur Verfügung (Abbildung 4.73). Ähnlich dem Table View stellt es die Daten tabellarisch dar — jedoch immer nur die zwei Dimensionen einer Spalte.

Kurzbeschreibung	
Verwendungszweck	Tabellenansicht zweidimensionaler Spalten
Eingang	Beliebige Tabelle mit beliebig vielen zweidimensionalen Spalten
Ausgabe	-
Besonderheit	-

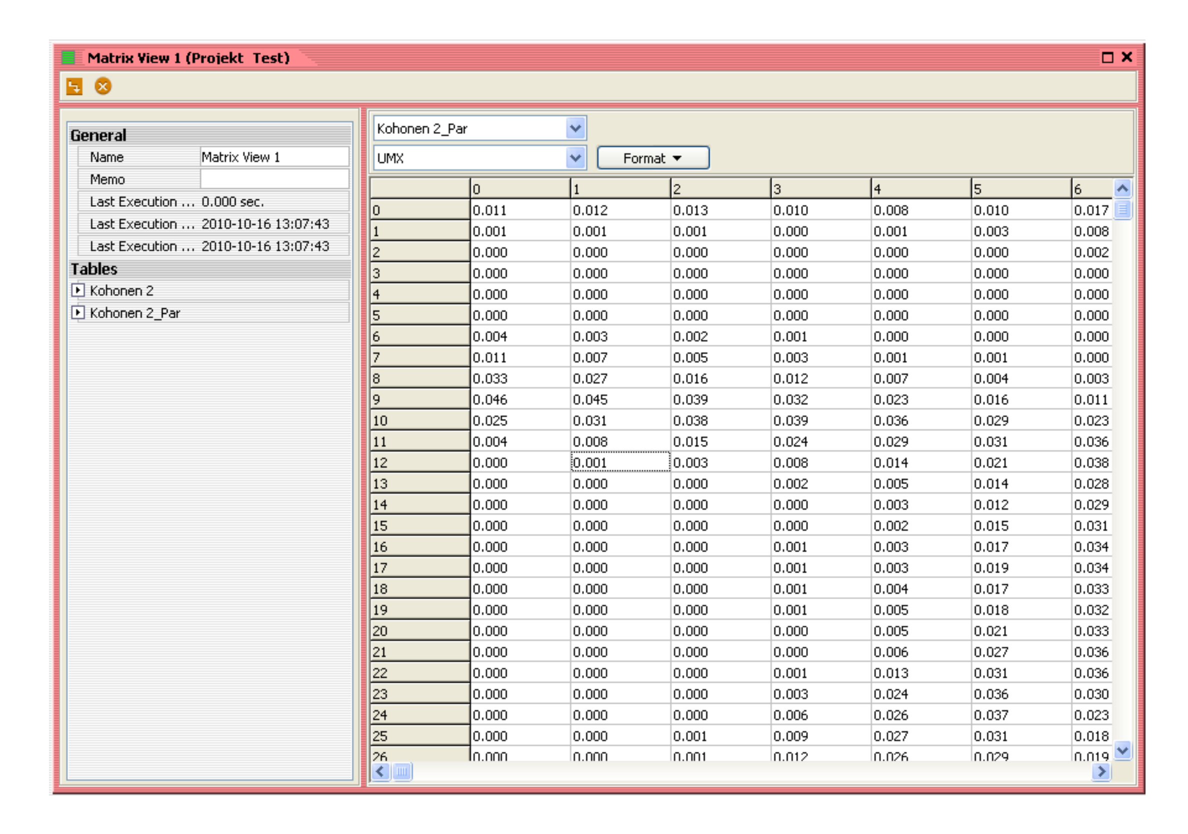


Abbildung 4.73.: Matrix–View

Wählen Sie in der oberen Auswahlbox die gewünschte Tabelle, in der unteren die darzustellende Spalte. Mit der Schaltfläche Format lässt sich analog zum Table View die Formatierung der anzuzeigenden Daten ändern (siehe 4.8.1 Formatierung ändern auf Seite 198). Matrizen, also zweidimensionale Spalten, werden unter anderem von den Modulen Kohonen (siehe 4.4.1, Seite 171) und Correlation (siehe 4.3.5, Seite 108) erzeugt.

4.8.3. Graphic View

Kurzbeschreibung	
Verwendungszweck	Graphische 2D- und 3D–Ausgabe
Eingang	Beliebige Tabelle mit beliebig vielen Spalten
Ausgabe	-
Besonderheit	-

Ein Bild sagt mehr als tausend Zahlen. Die grafische Darstellung (Abbildung 4.74) einer Analyse ist eines der leistungsfähigsten Werkzeuge des RayQ–Systems.

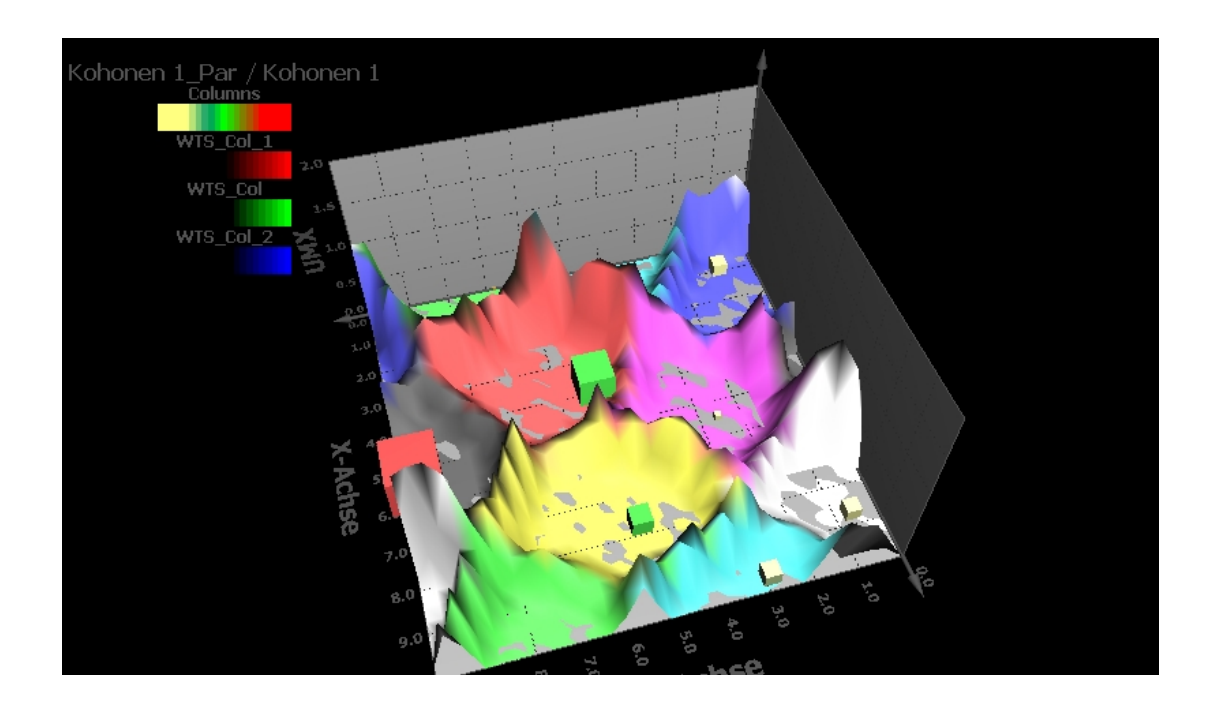


Abbildung 4.74.: Graphic–View

Der Graphic–View kann verschiedene Arten von Graphen darstellen. Ein Graph visualisiert eine Auswahl bestimmter Spalten einer Tabelle. Eine Kombination aus verschiedenen Tabellen in einem einzigen Graphen ist nicht immer möglich. Ein Graphic–View

kann jedoch mehrere Graphen in einem Bild gleichzeitig darstellen und somit auch aus verschiedenen Tabellen.

Disable Maxpoints

Um eine flüssige Darstellung der Grafiken auch auf langsameren Grafikkarten zu ermöglichen, werden im Normalfall nur die ersten 50000 Zeilen einer Tabelle zur Berechnung der Grafik herangezogen. Um diese Einschränkung zu umgehen, muss in der Windows–registry unter HKEY_CURRENT_USER\SOFTWARE\QYTE\RAYQ\ ein Schlüssel *DisableMaxPoints* vom Typ DWORD angelegt und diesem ein Wert ungleich 0 zugewiesen werden.

Neue Graphen einbinden

Um einen neuen Graphen in die Ansicht einzufügen, wählen Sie den gewünschten Typen über die Schaltfläche *Hinzufügen* aus. Ein neuer Graph ist eingerichtet, dem allerding zu dem jetzigen Zeitpunkt noch keine Spalte zugewiesen ist und der somit im Bild nicht sichtbar ist.

Tipp!
Ziehen Sie eine Spalte in eine leere Ansicht, schlägt Ihnen RayQ automatisch das Anlegen eines geeigneten Graphen vor.

Tipp!
Während Sie mit der Maus eine Spalte über den Graphen ziehen, wird die Achse rot eingefärbt, der Sie die Spalte zuweisen können.

Der neu angelegte Graph lässt sich komfortabel per Maus konfigurieren: Ziehen Sie mit der Maus eine Spalte der gewünschten Tabelle aus dem Modul–Inspektor in den Graphic View und legen Sie sie im Bild auf der Achse der Skala ab, auf der die Werte aufgetragen werden sollen. Abhängig vom Graphen–Typ lassen sich auf diese Weise auch weitere Einstellungen vornehmen. Enthält die Ansicht mehr als einen Graphen, so beziehen sich die Maus–Operationen ausschließlich auf den aktiven Graphen, der in der Auswahlbox (Abbildung 4.74) angezeigt wird.

Hinweis!
Bitte beachten Sie, dass gültige Daten in der betreffenden Tabelle vorliegen müssen, um einen Graphen darstellen zu können. Dazu ist in der Regel zuvor eine Berechnung der Analyse notwendig.

Die Skalierung stellt sich automatisch auf die angezeigten Graphen ein, um den zur Verfügung stehenden Platz vollständig ausnutzen zu können.

Bedienung

Die Bedienung des Graphic–Views erfolgt weitestgehend mit der Maus. Dabei wird zwischen dem reinen Ansichts–Modus (Bedienung siehe Tabelle 4.26) und dem Selektions–Modus, bei dem sich einzelne Datensätze markieren lassen, unterschieden (Bedienung siehe Tabelle 4.27). Die beiden Modi lassen sich über die Schaltflächen View und Selection am oberen Fensterrand auswählen.

Gedrückte Taste	Funktion
Linke Maustaste	Blickwinkel verändern
Strg-Taste + linke Maustaste	Zoom der Ansicht

Rechte Maustaste	Graph auf den horizontalen Ebenen verschieben (Kameraposition)
Strg-Taste + rechte Maustaste	Graph auf der vertikalen Ebene verschieben (Kameraposition)
Umschalt-Taste + rechte Maustaste	Verschiebung des Graphen (Kameraposition) auf Normalstellung zurücksetzen

Tabelle 4.26.: Ansicht steuern — View-Modus

Gedrückte Taste	Bezeichnung
Linke Maustaste	Blickwinkel verändern
Strg-Taste + linke Maustaste	Zoom der Ansicht
Rechte Maustaste	Selektionsrahmen auf den horizontalen Ebenen verschieben. Bei ausgeschaltetem Selektionsrahmen lassen sich mit der rechten Maustaste einzelne Datensätze durch Anklicken selektierenden oder deselektieren.
Strg-Taste + rechte Maustaste	Selektionsrahmen auf der vertikalen Ebene verschieben. Bei ausgeschaltetem Selektionsrahmen wird durch Klicken auf einen Datensatz die Selektion aller anderen Datensätze gelöscht.
Umschalt-Taste + rechte Maustaste	Selektionsrahmen auf den horizontalen Ebenen in der Größe verändern
Umschalt-Taste + Strg-Taste + rechte Maustaste	Selektionsrahmen auf der vertikalen Ebene in der Größe verändern

Tabelle 4.27.: Ansicht steuern — Selektions-Modus

Klicken Sie im View–Modus mit der Maus auf einen Graphen, um ihn als aktiv auszuwählen.

Legende verschieben. Mit gedrückter linker Maustaste lässt sich die Legende an eine beliebige Stelle innerhalb der Anzeige verschieben.

Graphen konfigurieren

Im Modul–Inspektor werden alle angelegten Graphen mit ihren jeweiligen Einstellungsmöglichkeiten aufgeführt, die Werkzeugleiste über dem Graphic–View stellt Ihnen allgemeine Möglichkeiten zur Konfiguration der Ansicht zur Verfügung (Tabelle 4.28).

Bezeichnung	Funktion
Hinzufügen	Fügt einen neuen Graphen hinzu.
Löschen	Löscht den aktuellen Graphen.
Umbenennen	Öffnet eine Dialogbox zum Ändern der Bezeichnung.
Zeichenmodus	Ermöglicht die Änderung der Darstellung (siehe jeweiligen Graphentyp).
Bild speichern	Speichert die aktuelle Ansicht als BMP oder JPG.
3D Modus	Die Anzeige lässt sich zwei- und dreidimensional darstellen. Eine 2D–Ansicht eignet sich zum Beispiel für einfache Verlaufskurven in Präsentationen, während 3D mehrdimensionale Graphen darstellen kann.
Aktiven Graphen hervorheben	Der in der Auswahlbox selektierte Graph wird in der Anzeige hell hervorgehoben, wenn diese Option gesetzt ist.
Wände	Stellt die Skalen als „graue Wände“ dar.
Legende	Legende ein- und ausblenden
Demo	Schaltet den Demonstrations–Modus ein und aus, der die Grafik um die Y–Achse dreht.
Sichtbar	Blendet den selektierten Graphen ein und aus.

Zoom	Stellt die Vergrößerung der Darstellung ein.

Tabelle 4.28.: Graphen konfigurieren (View–Modus)

Weitere Einstellungsmöglichkeiten stehen Ihnen im Modul – Inspektor zur Verfügung, die jedoch von dem jeweiligen Graphen–Typ abhängen.

Color Gradient

Der Dialog Color Gradient dient zum Festlegen von Farben und Farbverläufen in den Einstellungen des Graphic–View. Der Dialog (Abbildung 4.75) zeigt Ihnen eine schematische Darstellung, die Sie mit der Maus durch Ziehen der am oberen Rand angesetzten Schieber verändern können. Klicken Sie in den freien Raum zwischen oder neben die Schieber, wird ein neuer Punkt hinzugefügt. Ziehen Sie einen dieser Punkte mit der Maus aus den Bereich des Fensters nach oben oder unten „heraus", wird er aus dem Verlauf wieder entfernt. Statt dessen können Sie auch mit den Schaltflächen *Hinzufügen* einen neuen Punkt hinzufügen (der selektierte wird verdoppelt) und mit *Entfernen* den selektierten Punkt entfernen. Mit *Vorheriges* und *Nächstes* wechseln sie zwischen den einzelnen Punkten.

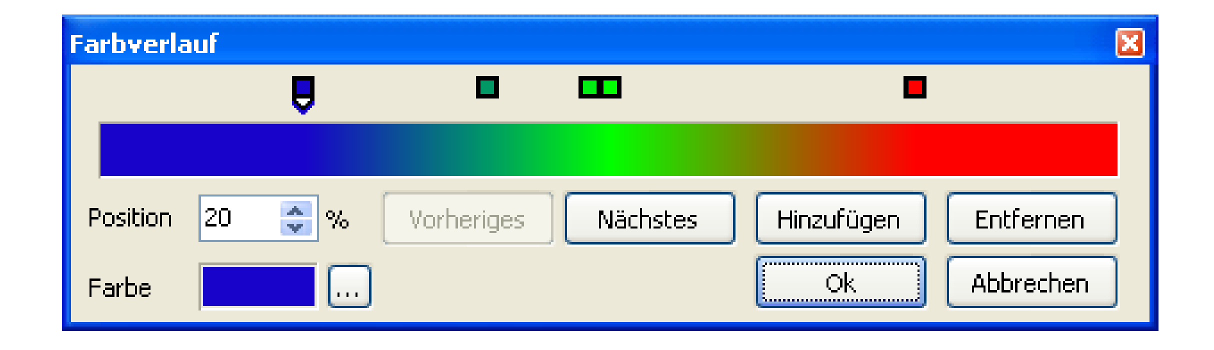


Abbildung 4.75.: Color–Editor

Klicken Sie auf den Button rechts neben dem Farbfeld am unteren Rand, um den Farbwert des zuletzt per Maus ausgewählten

Schiebers (erkennbar an dem kleinen Dreieck) zu verändern. Zwischen diesen Punkten werden die Farbverläufe interpoliert.

Grafik speichern

Die aktuelle Ansicht lässt sich jederzeit als Bitmap- (BMP) oder JPEG–Grafikdatei (JPG) abspeichern. Klicken Sie dazu auf *Bild speichern* und geben in dem sich öffnenden Dialog Speicherort, Dateiname und -typ entsprechend des gewünschten Datentyps an. Bei JPEG–Grafiken wird zusätzlich die Qualitätsstufe zur Speicherung abgefragt, wobei „100“ die höchste Qualität bezeichnet.

Markieren und Auswählen

Innerhalb eines Views wird zwischen einer Markierung und einer Selektion unterschieden: Markierungen werden in der jeweiligen RayQ–Tabelle eingetragen und stehen in allen anderen Modulen (z.B. Table–View) zur Verfügung. Eine Selektion dient zur Auswahl der zu markierenden Datensätze. Sie selektieren nacheinander beliebige Punkte im View und tragen diese Auswahl als Markierung über die Schaltfläche *Markierung hinzu* in die Tabelle ein (Tabelle 4.29).

Bezeichnung	**Funktion**
Markierung hinzu	Fügt eine Markierung auf die in der Ansicht selektierten Datensätz hinzu.
Markierung ersetzen	Ersetzt die in der Tabelle vorhandenen Markierungen durch Ihre Selektion.
Markierung entfernen	Entfernt eine Markierung auf den selektierten Datensätzen.
Markierung löschen	Löscht alle Markierungen der Tabelle.
Auswahl löschen	Setzt die Selektion in der Ansicht zurück.

Auswahlbox einblenden	Blendet einen würfelförmigen Selektionsrahmen in die Ansicht ein. Alle innerhalb dieses Rahmens befindlichen Datensätze werden selektiert. Über die Schieberegler lassen sich Position und Größe des Selektionsrahmens bestimmen.

Tabelle 4.29.: Selektions–Modus

Der mit *Auswahlbox einblenden* kann Ihnen bei der Selektion behilflich sein. Verändern Sie seine Position und Größe wahlweise über die drei Schieberegler oder direkt mit der Maus innerhalb der Ansicht, bis sich die zu selektierenden Datenpunkte innerhalb des Rahmens befinden.

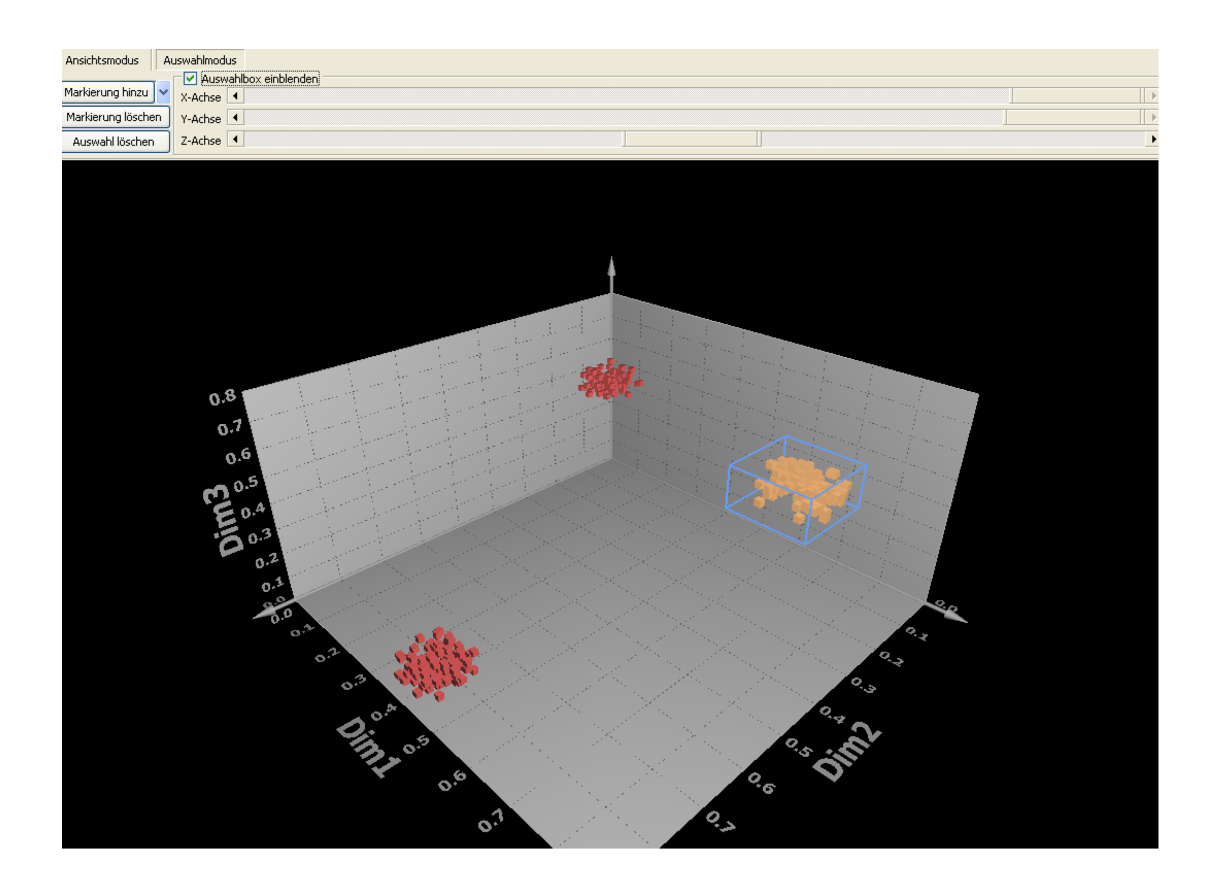


Abbildung 4.76.: Selektions–Modus

Scatter

Ein „Scatter“ ist eine reine Auftragung der Eingangsdaten auf die gewählten Achsen als Datenpunkte. Der Graphic–View kann ein– bis dreidimensionale Scatter darstellen. Dabei wird jeder Datensatz der Tabelle als Würfel (3D) oder Quadrat (2D) in den Graphen gezeichnet. Die einzelnen Datenpunkte lassen sich mit einer Linie verbinden — wahlweise als Gerade *(Linie)* oder Bezier–Kurve *(Spline)*, um den Verlauf mit interpolierten Zwischenpunkten Welche Einstellungsmöglichkeiten Sie im Detail vornehmen können, lesen Sie in Tabelle 4.30.

Bezeichnung	**Funktion**
Drawmode	Ändert die Darstellung des Graphen Linien: Stellt nur die direkten Verbindungslinien zwischen den Datenpunkten dar. Linien mit Würfeln: Stellt die direkten Verbindungslinien gemeinsam mit den Datenpunkten dar. Kurven: Stellt den Graphen als durchgehende Kurve dar. Kurven mit Würfeln: Zu der durchgehenden Kurve werden die Datenpunkte eingeblendet. Kurven: Stellt nur die Datenpunkte in Form von Würfeln/Quadraten ohne Verbindungslinien dar.
Line style	Verändert die Linienart, zur Auswahl stehen: durchgezogen, gepunktet, Gepunktet (weit), Streifen, Streifen (weit)
Line size	Linienstärke
Base object size	Größe der Datenpunkte
X–Axis, Y–Axis, Z–Axis	Gibt die Spalte der Tabelle an, die auf der jeweiligen Achse aufgetragen werden soll
Object size	Variiert die Größe der Datenpunkte relativ zu einer weiteren Spalte der Tabelle.

Color	Variiert die Farbe der Datenpunkte relativ zu einer weiteren Spalte der Tabelle.

Tabelle 4.30.: Scatter Parameter

Hinweis!
Die Interpolation erfolgt durch den OpenGL–Treiber und darf daher nur mit entsprechender Rücksicht auf dessen Rechengenauigkeit verwendet werden. Die Verwendung dieser Option ist nur bei wenigen Datenpunkten sinnvoll, da die Berechnung der interpolierten Zwischenpunkten zusätzliche Rechenzeit beansprucht.

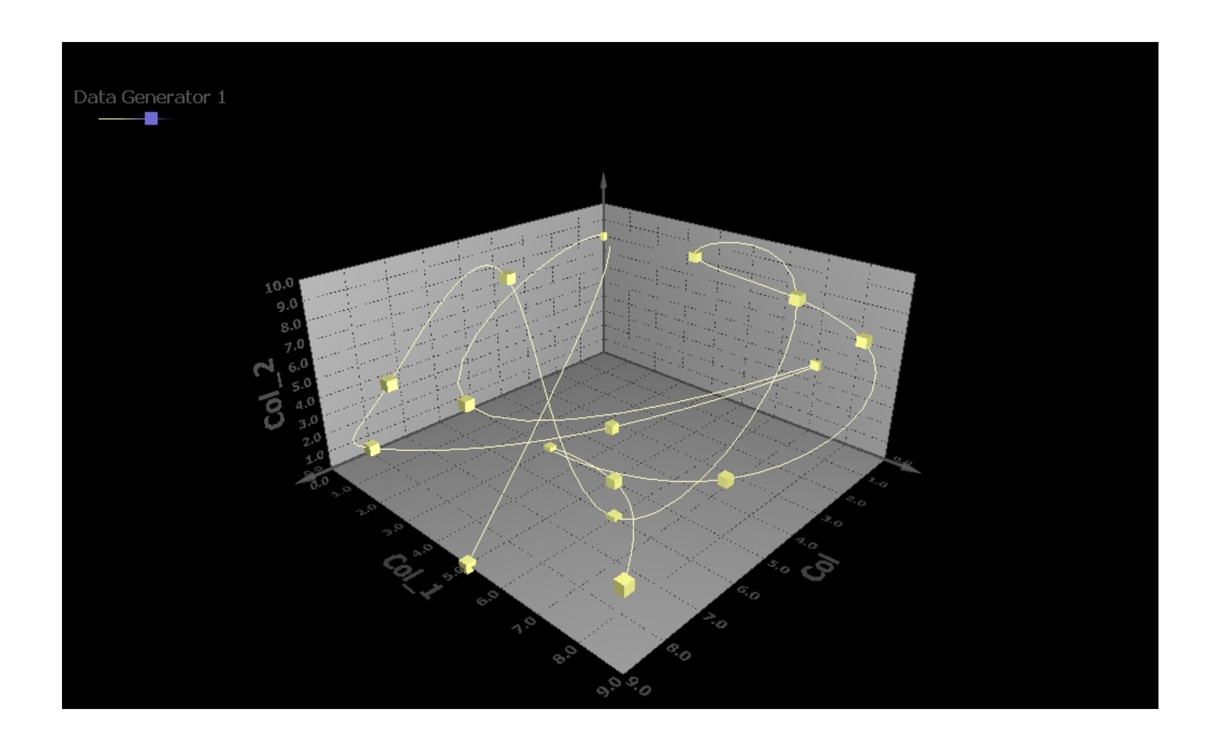


Abbildung 4.77.: Scatter 3D

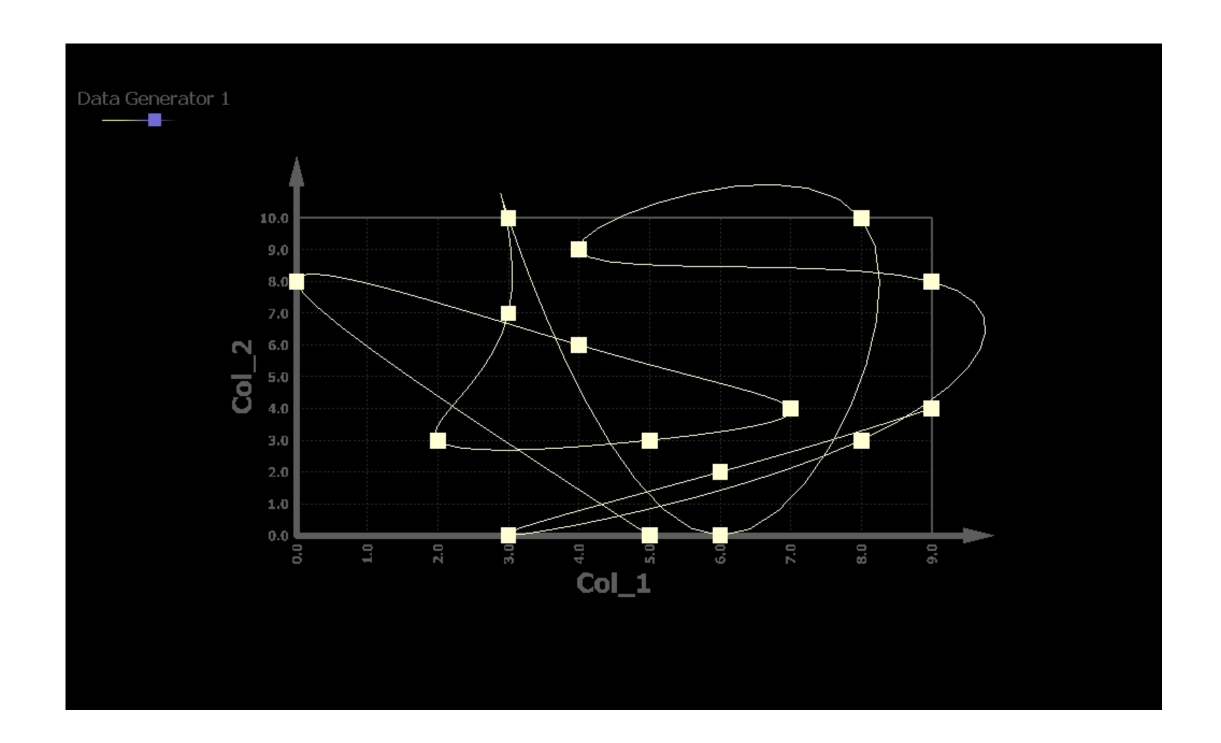


Abbildung 4.78.: Scatter 2D

Surface

Tabellen können neben eindimensionalen Spalten auch zweidimensionale Matrizen enthalten. Sie sind zum Beispiel das Ergebnis einer Kohonen–Berechnung (Siehe 4.4.1 Kohonen auf Seite 171.). Die Matrix wird mithilfe eines Oberflächendiagramms dargestellt, das die einzelnen Punkte horizontal mit einer Fläche „überspannt“ (Abbildung 4.79). Ziehen Sie zu diesem Zweck die gewünschte Spalte der Tabelle auf eine beliebige Skala des Graphic Views.

Eine bestehende Oberfläche lässt sich zusätzlich mit weiteren zweidimensionalen Matrizen parametrisieren, die als gefärbte Bereiche dargestellt werden. Die betreffenden Spalten ziehen Sie mit der Maus auf die vorhandene Fläche. Es wird automatisch ein neuer Farbverlauf angelegt, der den Wertebereich auf der Oberfläche visualisiert.

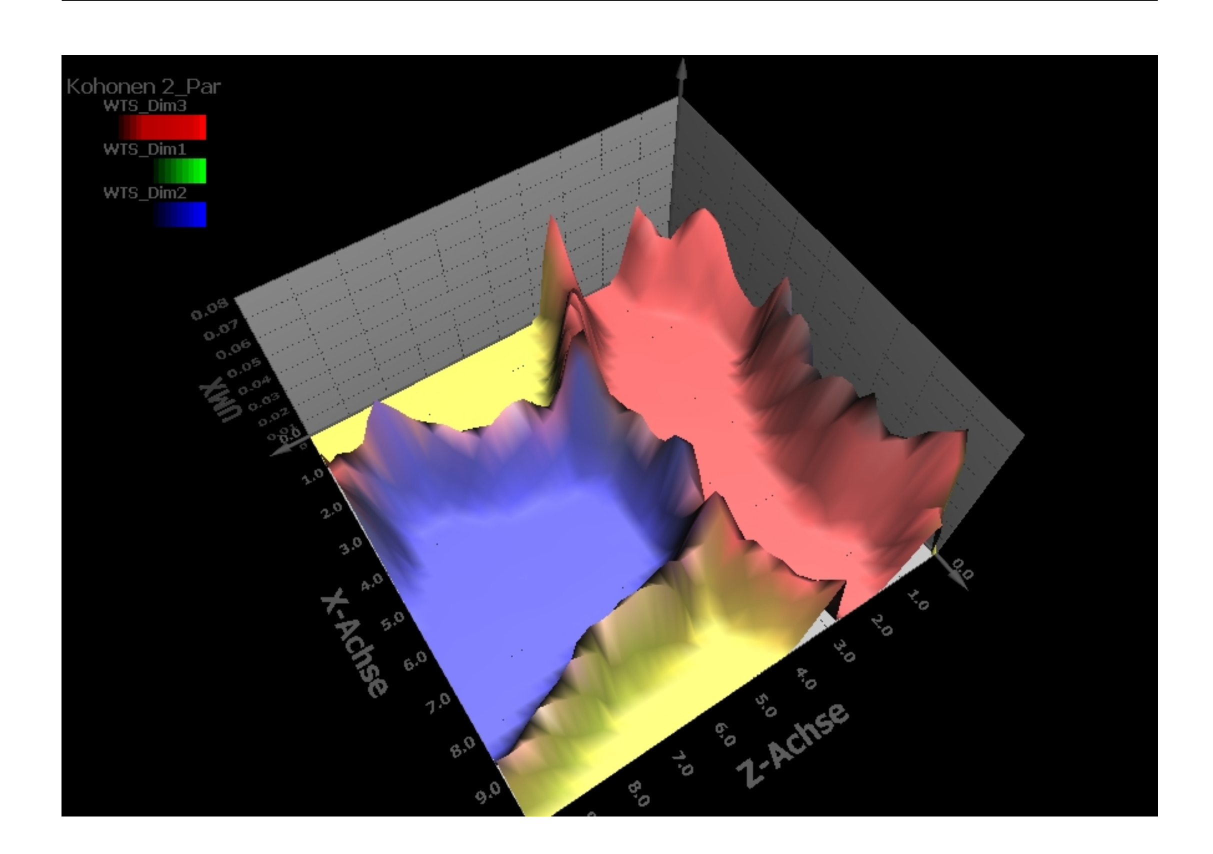


Abbildung 4.79.: Surface 3D

Zusätzlich zu den dreidimensionalen Daten lassen sich noch zwei eindimensionale Reihen hinzufügen, die wie bei einem Scatter als Würfel auf der Oberfläche dargestellt werden. Fallen mehrere Würfel auf einen Punkt, wird ein dementsprechend größerer Würfel gezeichnet. Das Verhältnis dieser Größenunterschiede lässt sich durch einen Minimal– und Maximalwert in den Parametern vorgeben. Diese Darstellung ist für die Anzeige der „Best Matches“ vorgesehen, die vom Kohonen–Modul berechnet werden (siehe 4.4.1, Seite 171).

Die Bedeutung der einzelnen Parameter entnehmen Sie der Tabelle 4.31.

Bezeichnung	Funktion
Drawmode	Bestimmt die Darstellung der Oberfläche. Flächen: Stellt ein zusammenhängendes Oberflächendiagramm dar farbige Linien: Zeichnet ein der Oberfläche entsprechend eingefärbtes Drahtgittermodell, das nur die einzelnen Punkte der Matrix verbindet. weiße Linien: Zeichnet ein rein weißes Drahtgittermodell, das nur die einzelnen Punkte der Matrix verbindet. Flächen und Linien: Oberfläche und Gitter werden gleichzeitig dargestellt.
Flat Colors	Deaktiviert die Berechnung von Farb–Zwischenwerten
Shining	Fügt Lichteffekte in die Grafik ein
Show Topnormals	Zeigt die Flächennormalen an der Oberseite des Oberflächengraphen an
Show Bottomnormals	Zeigt die Flächennormalen an der Unterseite des Oberflächengraphen an
Normals Length	Bestimmt die Länge der Fächennormalen (sofern verwendet)
Base object size, min, max	Ausgangsgröße der Würfel (sofern verwendet). Je mehr Würfel auf einen Punkt fallen, desto größer die Darstellung im Bereich von *min* und *max*.
Y–Axis	Matrix zur Darstellung der Oberfläche
Color Gradient	Matrix zur Färbung der Oberfläche
Cube Columns	Spalte als Ausgangswert für die X–Achse der Würfel
Cube Rows	Spalte als Ausgangswert für die Z–Achse der Würfel
Cube Color Gradient	Spalte für die Färbung der Würfel

Tabelle 4.31.: Surface Parameter

Box–Plot

Box–Plots dienen der übersichtlichen grafischen Darstellung von Beobachtungsreihen. Anhand eines Box–Plots lassen sich Verteilung und Struktur der Datenreihe einfach ablesen.

Bezeichnung	Funktion
Drawmode	Bestimmen Sie die Darstellungsart, z.B. transparent oder nur als Linien
Body Color	Farbe der Box und der Inner Fences
Data Color	Farbe des Medians und der Outer Fences
X–Axis, Y–Axis, Z–Axis	Bestimmt, an welcher Achse der Box–Plot aufgetragen werden soll.

Tabelle 4.32.: Boxplot Parameter

Der h–spread (Abbildung 4.80, (2)) enthält 50% aller Beobachtungswerte. Die Werte zwischen *inner-* (4) und *outer fence* (5) bilden Ausreißer. Die Werte außerhalb der outer–fences sind so genannte krasse Ausreißer. Ist der Box–Plot asymmetrisch, so ist auch die Verteilung der Beobachtungsreihe asymmetrisch.

Der Box–Plot–Graph bezieht seine Daten aus einem Boxplot Calculation Modul (siehe 4.3.3, Seite 101), das zwischen Datenquelle und Graphic View geschaltet sein muss. Setzen Sie andere Tabellen als Eingangsdatenmengen ein, wird ein Box–Plot–Graph angezeigt, der jedoch keinerlei Aussage besitzt.

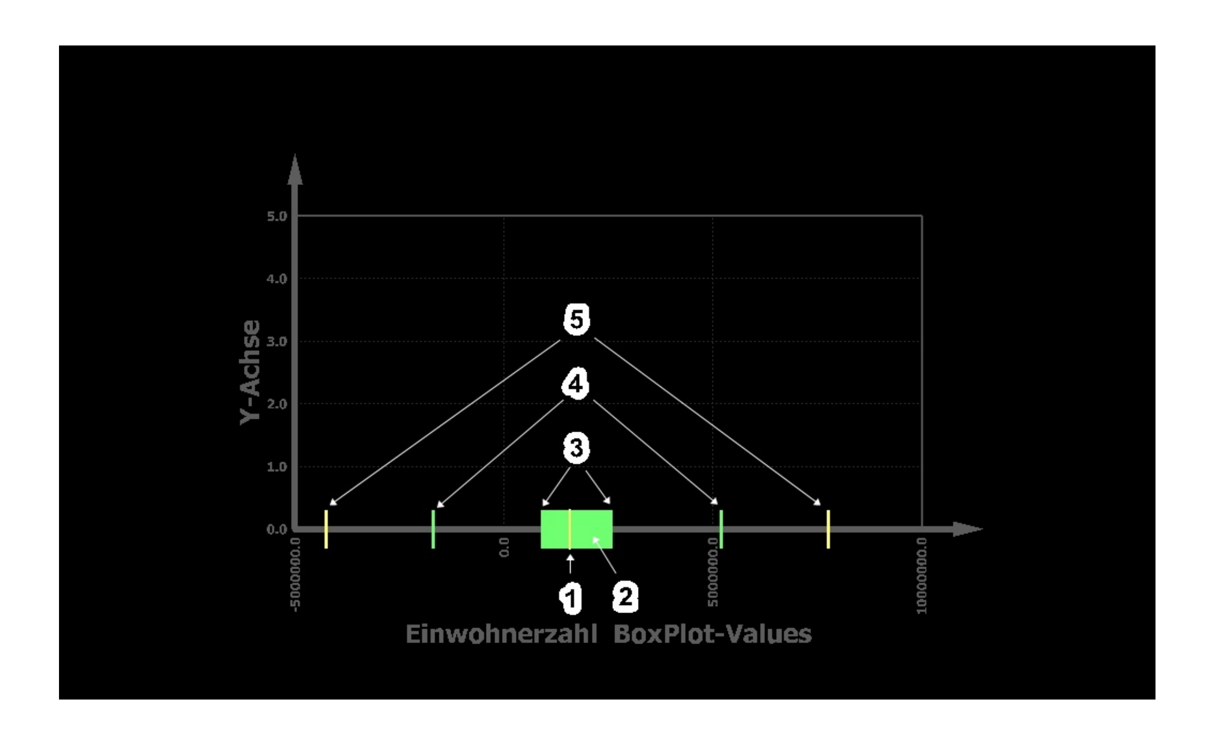


Abbildung 4.80.: Box–Plot

1. Median
2. „Box“ (h–spread)
3. Hinges
4. Inner Fence
5. Outer Fence

Histogramm

Das Histogramm trägt ähnlich dem Scatter Daten auf zwei Achsen auf, jedoch als Balkendiagramm. Als Besonderheit besitzt das Histogramm die Möglichkeit, Untergruppen darzustellen, die vom Group–Modul (siehe 4.3.9, Seite 125) geliefert werden.

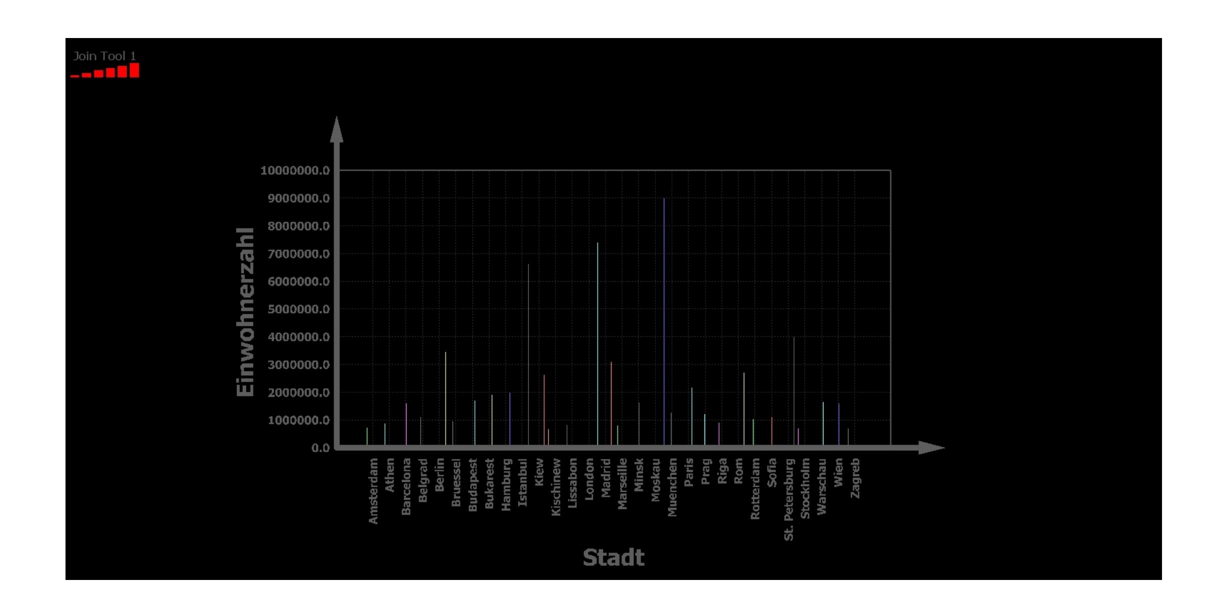


Abbildung 4.81.: Histogramm

Bezeichnung	**Funktion**
Drawmode	Anzahl aller Einträge dieser Spalte
Group Color	Anzahl aller Einträge dieser Spalte ungleich NULL
Border Space	Summe der Werte
Group Space	Kleinster Wert
Subgroup	Größter Wert
Show as Scaling	Varianz der Werte
Data	Quadratsumme
Quad Avg	Quadratdurchschnitt

Tabelle 4.33.: Parameter Histogramm

Für einfache, ungruppierte Daten ziehen Sie die gewünschten Spalten wie bei Scatter auf die entsprechenden Achsen. Für die Gruppierung wählen Sie in der Parameter–Liste unter *Subgroup 1* bis *3* die gewünschte Spalte als Untergruppe aus. Der Parameter *Show as Scaling* gibt dabei jeweils an, welche der Gruppen als Beschriftung am unteren Skalenrand angezeigt werden soll.

KAPITEL 5

Administration

Zur Konfiguration des RayQ Servers wird die RayQ–Administrations – Software mitgeliefert. Sie dient dem Systemadministrator zum Verwalten der RayQ–Benutzer und zum Einrichten zur Verfügung stehenden Datenquellen. Zudem gibt die Administrations–Software einen Überblick über die laufenden Prozesse und die gespeicherten Projekte.

5.1. Start und Anmeldung

Nach Programmstart zeigt die Administrations–Software genauso wie der RayQ Designer einen Anmeldebildschirm, in den Sie den als Administrator berechtigten Benutzernamen und das dazugehörige Kennwort eingeben.

Wichtig!

Klicken Sie auf die Schaltfläche *Advanced*, um die korrekte IP–Adresse und den Port des Servers anzugeben. Die RayQ–Administration muss nicht auf demselben System gestartet werden, auf dem auch der Server ausgeführt wird. Eine Administration über das Netzwerk ist jederzeit möglich.

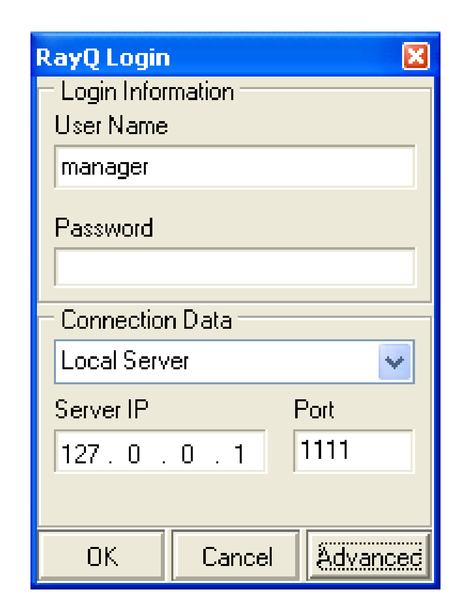


Abbildung 5.1.: Anmeldung an die RayQ–Administration mit geöffnetem Advanced–Bereich zur Eingabe der Server–Adresse

Hinweis!

Die Benutzerdaten des RayQ–Administrators müssen bei der ersten Installation des RayQ–Servers angegeben werden. Verwenden Sie bitte diese Daten, um sich das erste Mal bei der Administrations–Software anzumelden.

5.2. Hauptfenster

Das Hauptfenster der RayQ–Administration unterteilt sich in die folgenden fünf Bereiche:

5.2.1. Status

Das Status–Fenster informiert Sie über die aktuell angemeldeten Benutzer am RayQ–System. Die Spalte *Host* zeigt den Rechner, von dem aus der jeweilige Benutzer angemeldet ist. *Client Active* gibt an, ob zur Zeit ein RayQ–Designer auf dem betreffenden Rechner betrieben wird *(true)* oder nur ein Server–Prozess des Benutzers aktiv ist *(false)*. In *Active Sessions* steht die Anzahl der offenen Projekte des jeweiligen Benutzers.

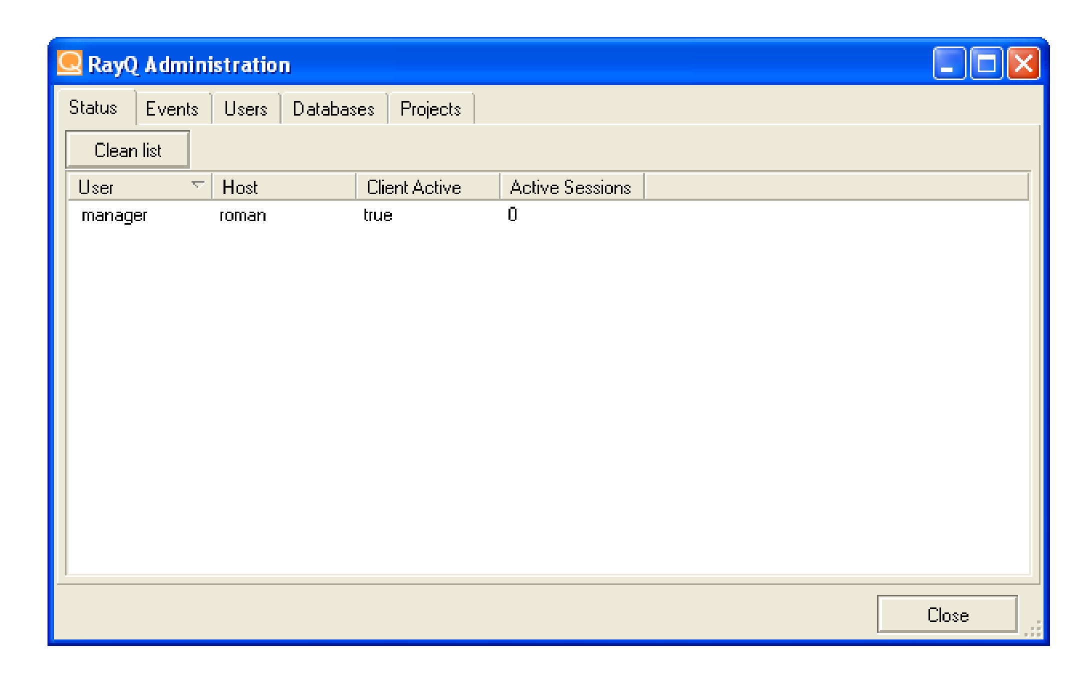


Abbildung 5.2.: Aktueller Status des RayQ–Systems

5.2.2. Events

Das Ereignisprotokoll (Abbildung 5.3) führt alle systemrelevanten Aktionen der einzelnen Benutzer auf. Die Anzeige der Einträge lässt sich über einen Filter einschränken, so dass entweder eine feste Menge oder die Meldungen eines bestimmten Zeitraums angezeigt werden. Klicken Sie auf *Configure*, um die Einstellungen

des Anzeigefilters zu verändern (siehe Abbildung 5.4). Der Filter bezieht sich lediglich auf die Anzeige, die Protokollierung selbst wird uneingeschränkt durchgeführt.

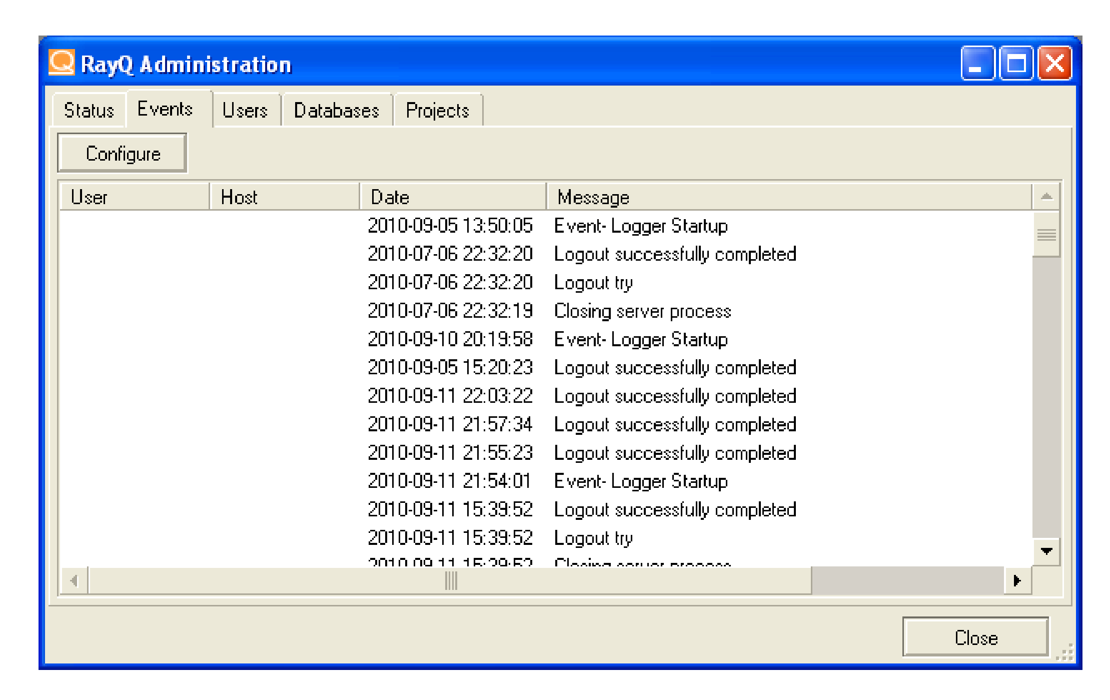


Abbildung 5.3.: Das Ereignisprotokoll zeigt die letzten Aktionen im RayQ–System

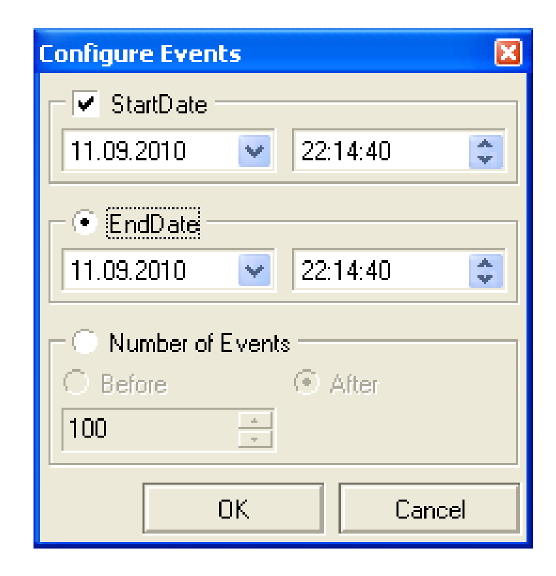


Abbildung 5.4.: Über einen einfachen Filter lässt sich die Anzeige des Protokolls einschränken

5.2.3. Users

Das RayQ–System erlaubt die Verwaltung einer beliebigen Anzahl Benutzer, die wahlweise auch Administrationsrechte zugewiesen bekommen können. Über die Schaltflächen *Add* fügen Sie neue Benutzer hinzu, mit *Remove* entfernen Sie sie und mit *Modify* können die Daten bestehender Benutzer angepasst werden.

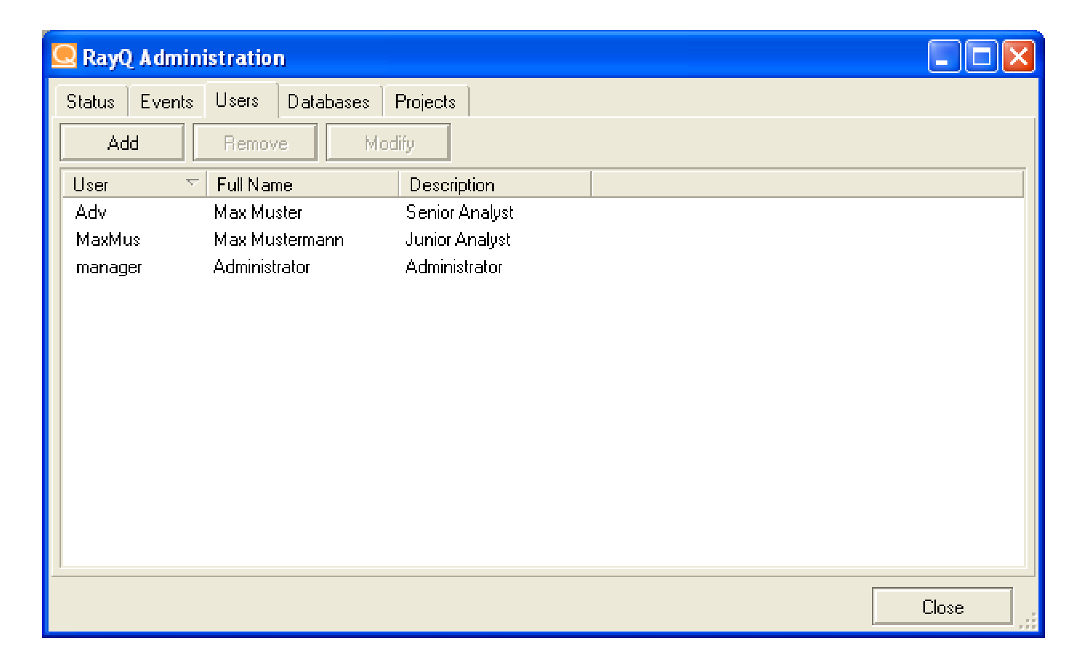


Abbildung 5.5.: Benutzer des RayQ–Systems

Klicken Sie auf *Add* oder *Modify*, öffnet sich der Dialog aus Abbildung 5.6. Vergeben Sie einen Namen *(Username)* und ein Kennwort *(Password)*, das der Benutzer bei der Anmeldung angeben muss. Die Angabe von *Full name* und *Description* sind optional und dienen nur zu Ihrer Information.

Ist die Option *User is Administrator* ausgewählt, hat der Benutzer als Administrator vollen Zugriff auf das RayQ–System und dessen Konfiguration.

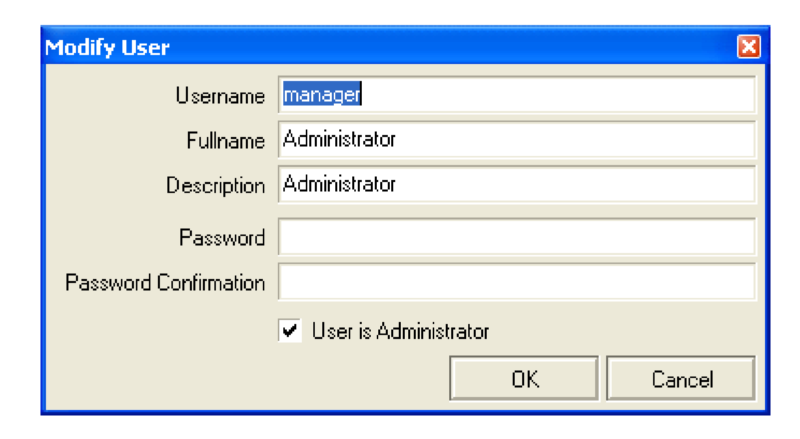


Abbildung 5.6.: Bearbeiten Sie die Einstellungen jedes Benutzers

5.2.4. Databases

Der RayQ Server unterstützt den direkten Zugriff auf ORACLE-Datenbanken und über die ODBC–Schnittstelle auf entsprechend unterstützte Systeme. Die Konfiguration der Datenbanken erfolgt über die Seite *Databases* (Abbildung 5.7).

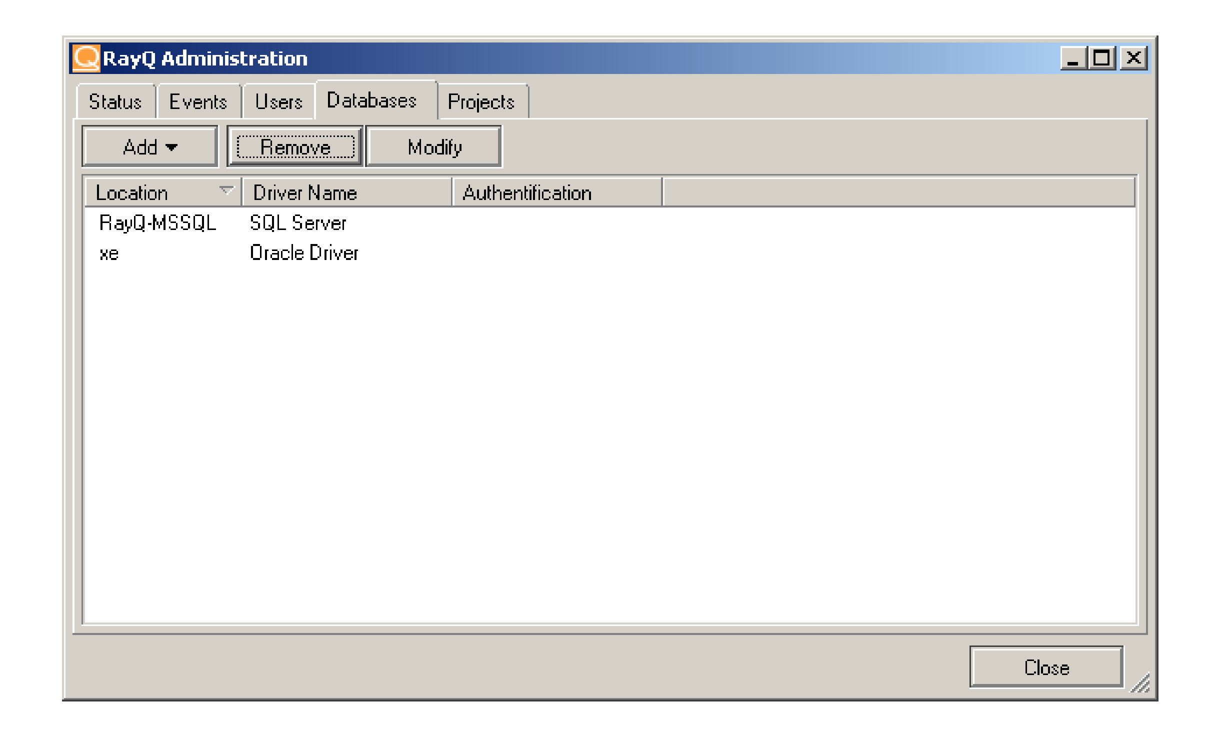


Abbildung 5.7.: Verwaltung der über den RayQ–Server erreichbaren Datenbanken

Hinweis!
Informationen über die Installation der Datenbanktreiber erhalten Sie ab Seite 10.

ORACLE

Für einen nativen Zugriff auf eine ORACLE–Datenbank muss ein ORACLE–Client auf dem Rechner installiert sein, auf dem der RayQ–Server betrieben wird.

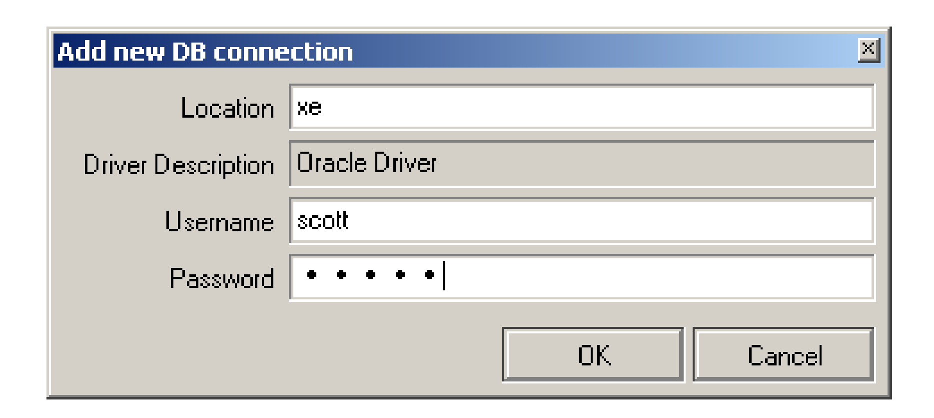


Abbildung 5.8.: Eingabe einer Verbindung zur einer ORACLE–Datenbank

Klicken Sie im RayQ Administrator auf der Seite Databases auf die Schaltfläche *Add* und wählen den Eintrag Oracle Driver aus. In dem sich öffnenden Eingabefenster (Abbildung 5.8) geben Sie unter Location die Bezeichnung des Dienstes ein, der im ORACLE–Client eingerichtet ist. Geben Sie zusätzlich Benutzername und Kennwort des Zugriffskontos ein — nach Bestätigung mit *Ok* wird die Verbindung getestet und in die Liste der Datenquellen im RayQ Server eingetragen.

> **Hinweis!**
> Der native ORACLE–Datenbankzugriff steht nur zur Verfügung, wenn der entsprechende Treiber bei der Installation des RayQ Server ausgewählt wurde.

ODBC

Der RayQ Server kann auf alle eingerichteten ODBC – Systemdatenquellen zugreifen. Klicken Sie auf die Schaltfläche *Add* und wählen den ODBC–Datenbanktreiber aus, den Sie verwenden möchten. In dem sich anschließend öffnenden Dialogfenster geben Sie den Namen der jeweiligen Systemdatenquelle an.

> **Hinweis!**
> Der ODBC–Zugriff steht Ihnen nur zur Verfügung, wenn der ODBC–Treiber bei der Installation des RayQ Servers ausgewählt wurde. Die von ODBC zur Verfügung gestellten Datenbank–Treiber sind von Ihrem eingesetzten System und den dort installierten Treibern abhängig.

ODBC — Einrichtung SQL–Server

Folgen Sie zum korrekten Einrichten eines MS SQL–Servers der Abbildungen 5.9 bis 5.15.

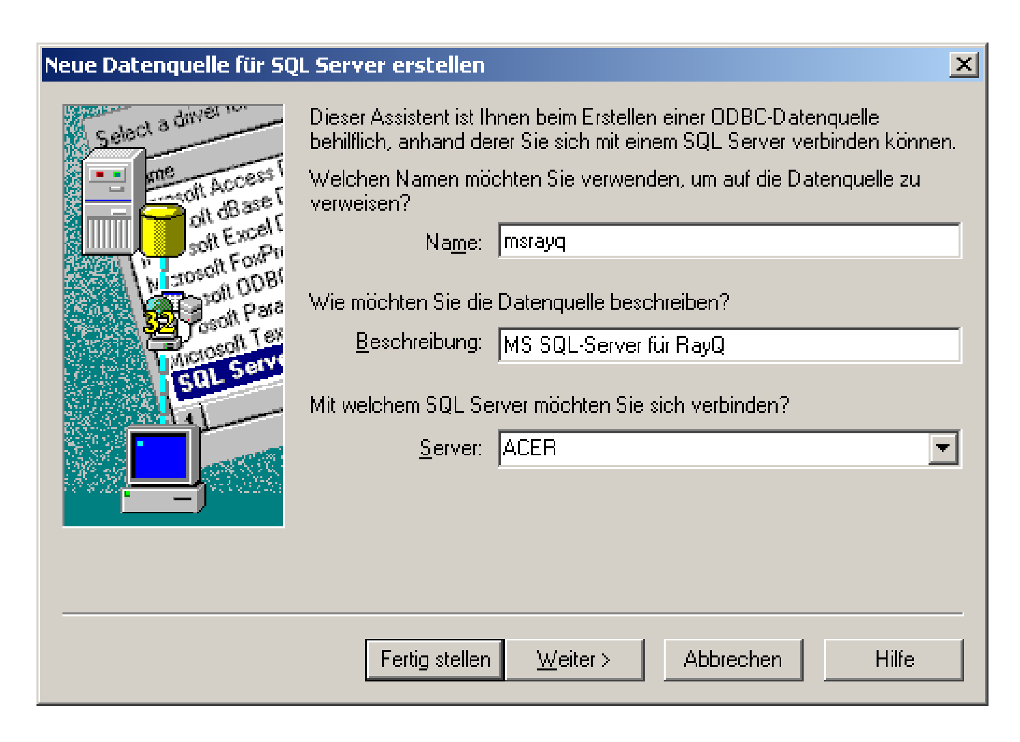


Abbildung 5.9.: Einrichtung SQL–Server: Schritt 1

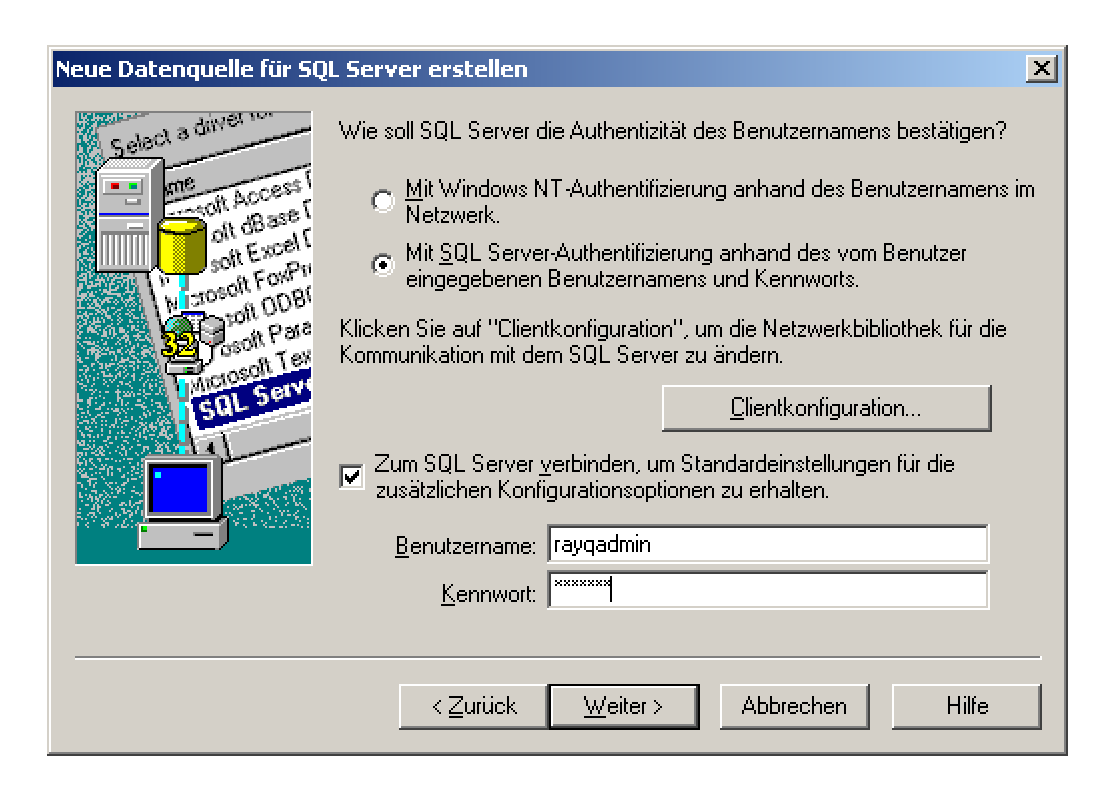


Abbildung 5.10.: Einrichtung SQL–Server: Schritt 2

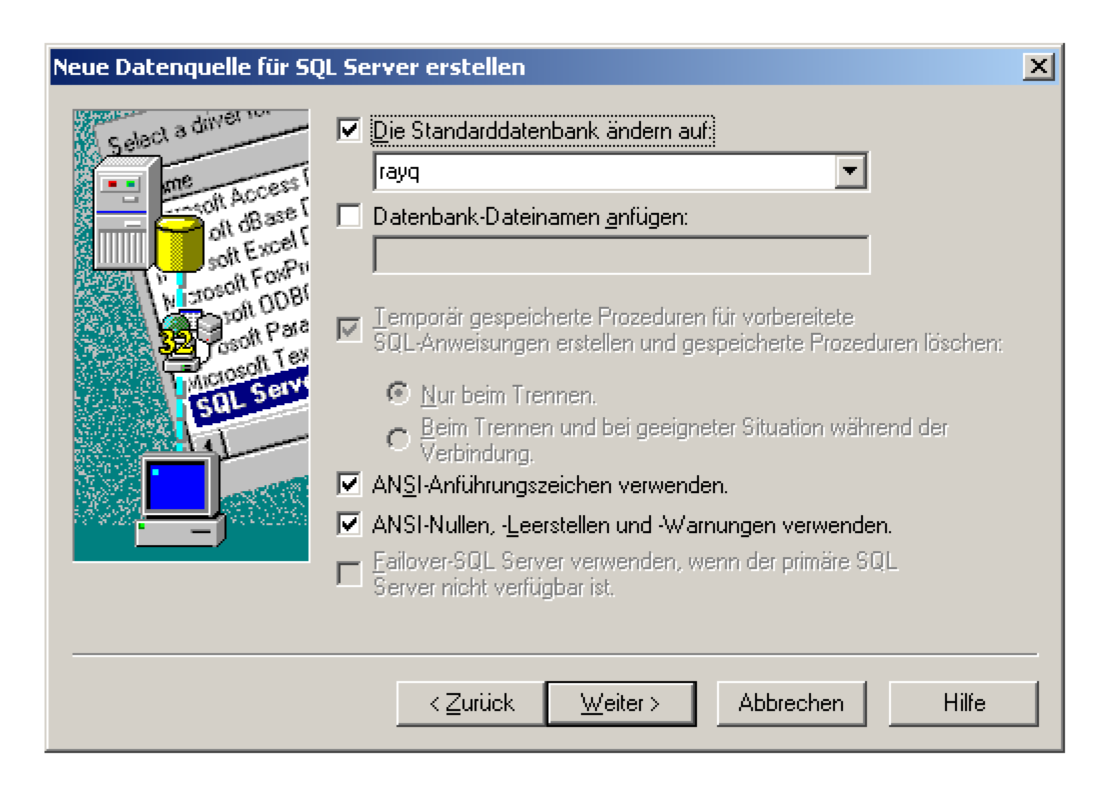


Abbildung 5.11.: Einrichtung SQL–Server: Schritt 3

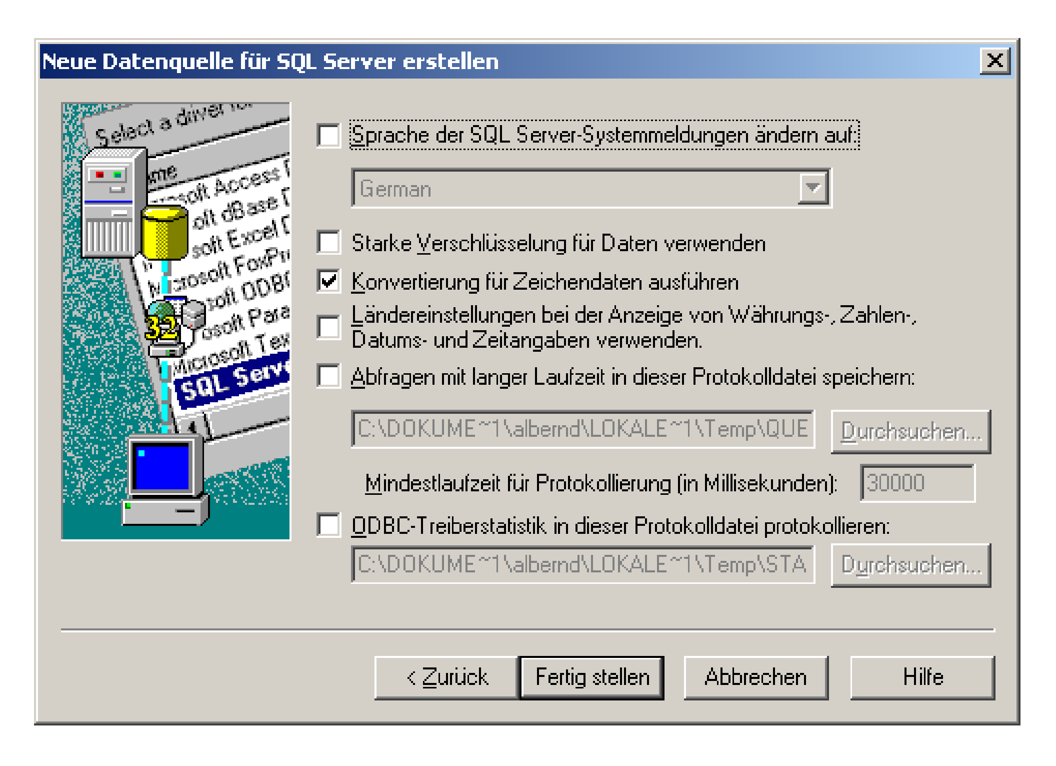


Abbildung 5.12.: Einrichtung SQL–Server: Schritt 4

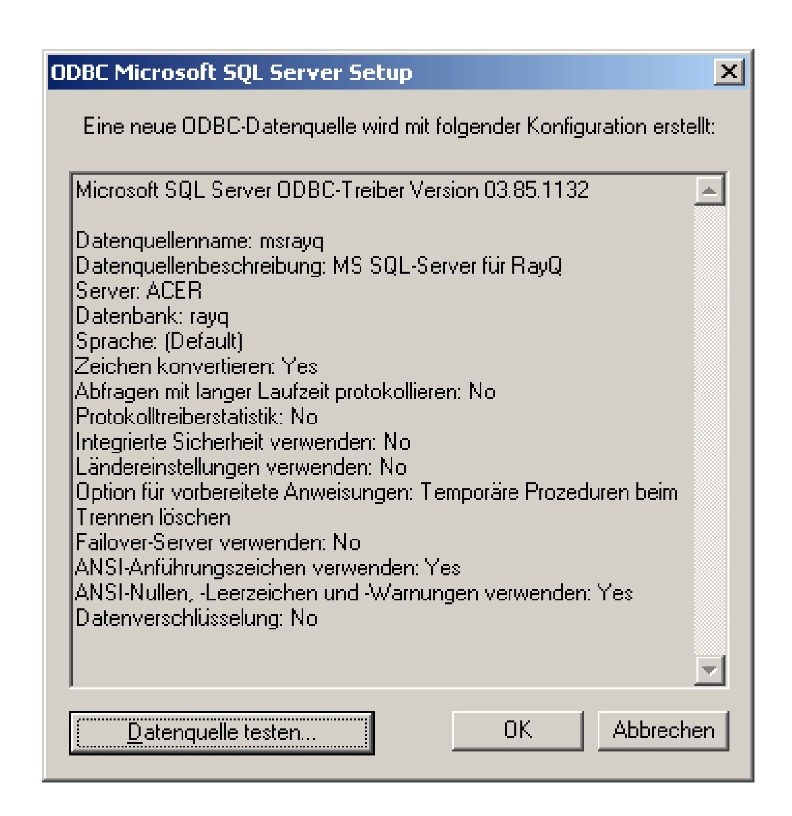


Abbildung 5.13.: Einrichtung SQL–Server: Schritt 5

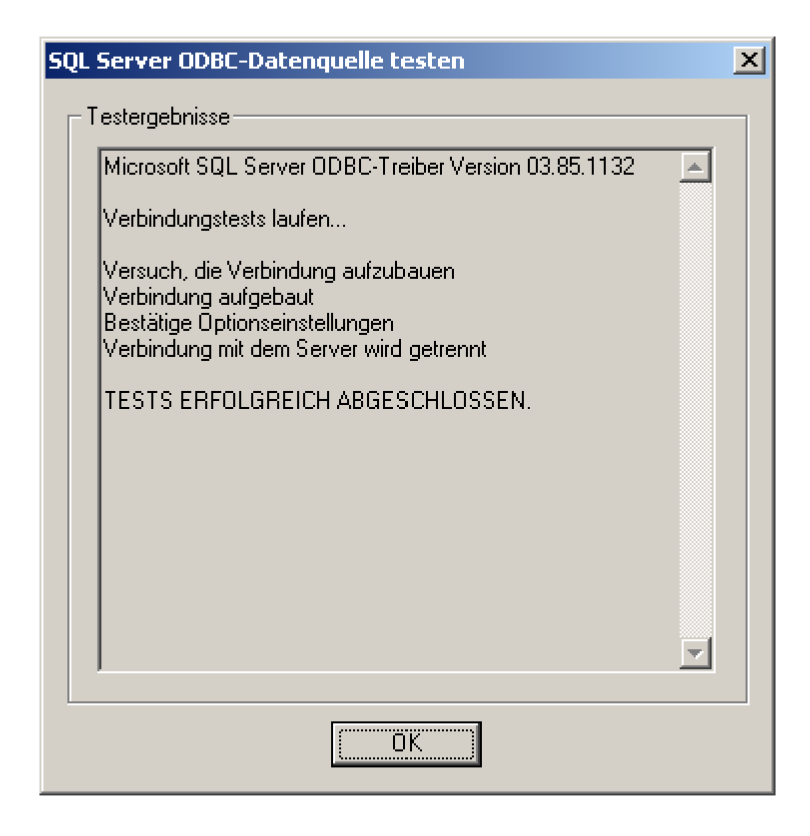


Abbildung 5.14.: Einrichtung SQL–Server: Datenquelle testen

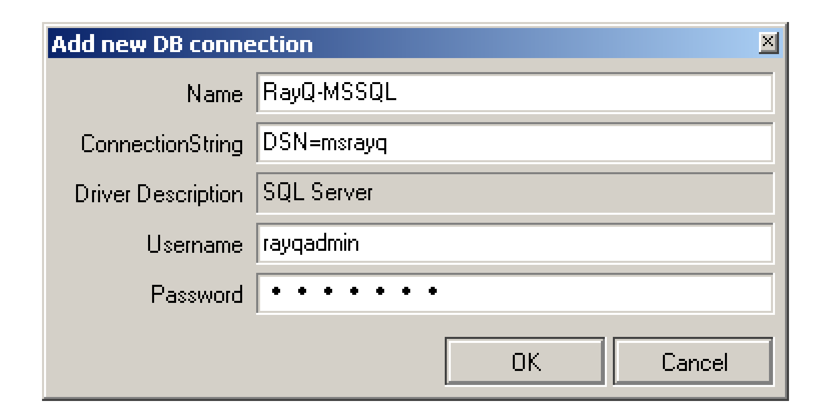


Abbildung 5.15.: Einrichtung SQL–Server: Integration in RayQ

ODBC — Einrichtung Sybase IQ

Folgen Sie zum korrekten Einrichten der Sybase IQ der Abbildungen 5.16 bis 5.20.

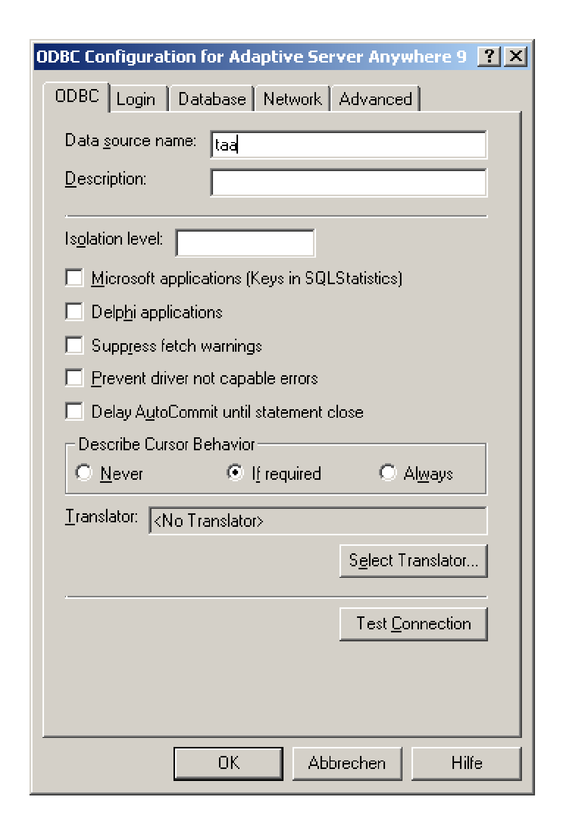


Abbildung 5.16.: Einrichtung Sybase IQ: Schritt 1

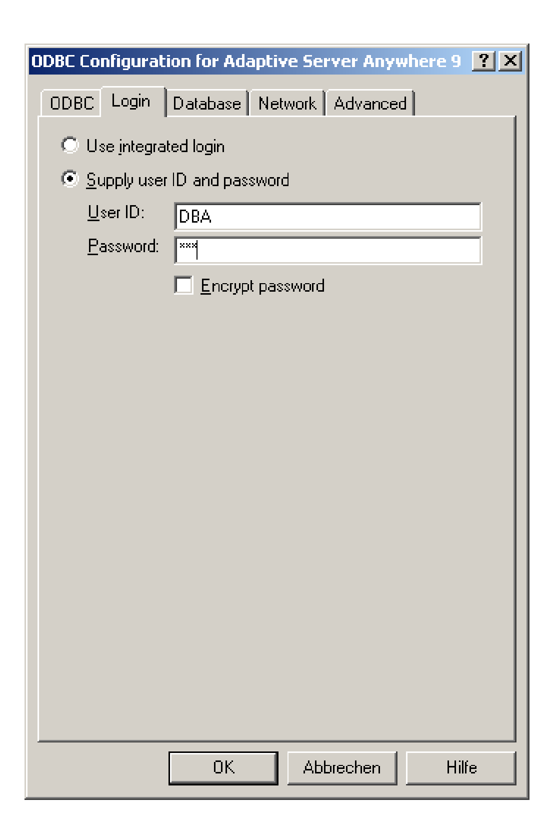


Abbildung 5.17.: Einrichtung Sybase IQ: Schritt 2

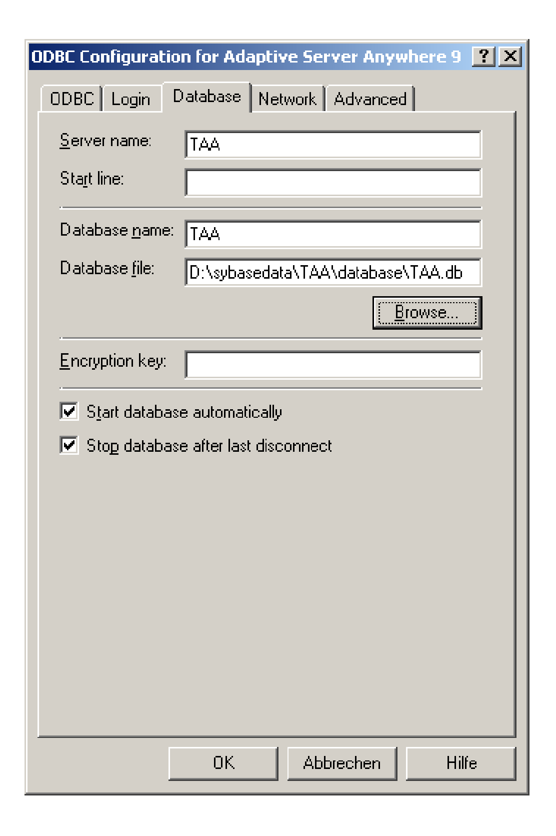


Abbildung 5.18.: Einrichtung Sybase IQ: Schritt 3

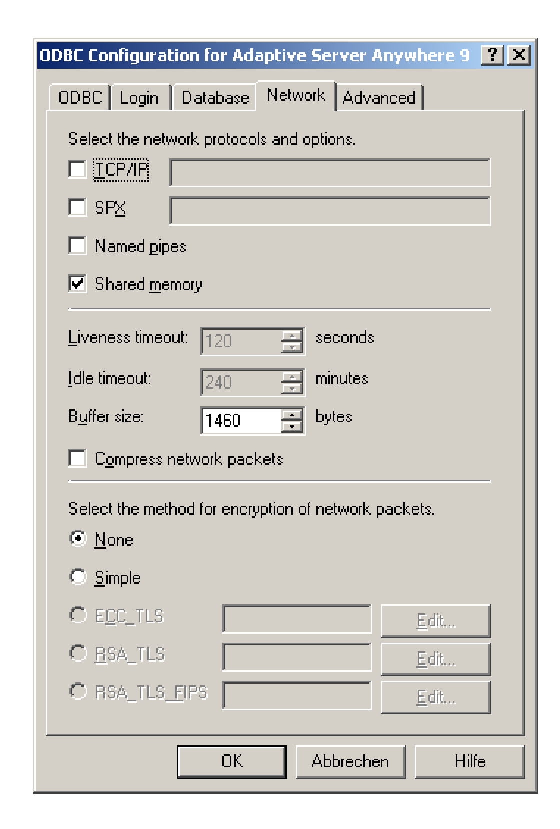


Abbildung 5.19.: Einrichtung Sybase IQ: Schritt 4

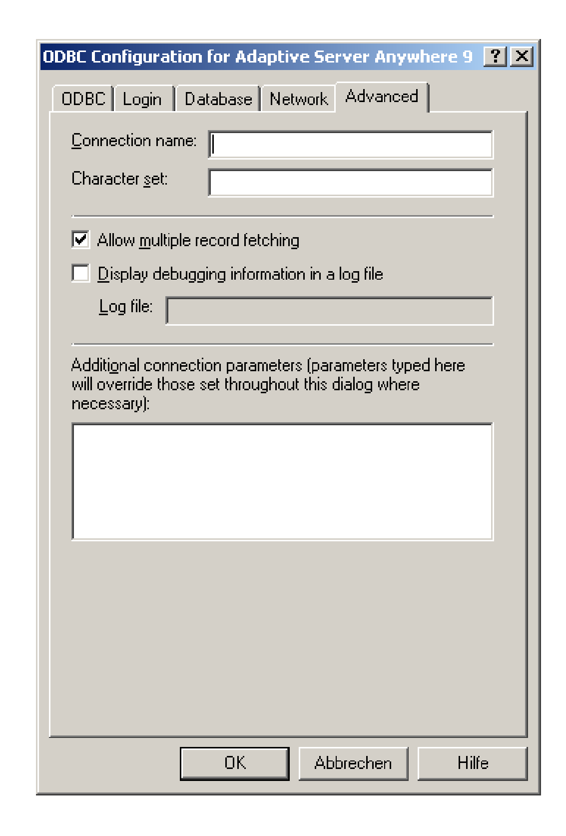


Abbildung 5.20.: Einrichtung Sybase IQ: Schritt 5

Datenbank löschen

Wählen Sie die Datenquelle, die Sie entfernen möchten, in der Liste aus und klicken auf die Schaltfläche Remove. Die Datenbank steht fortan im RayQ–System nicht mehr zur Verfügung.

> **Hinweis!**
> Bedenken Sie, dass das Entfernen einer Datenbank aus dem System Auswirkungen auf bestehende Analyseprojekte haben kann, die diese Datenquelle verwenden.

5.2.5. Projects

Auf der Seite *Projects* erhalten Sie eine Übersicht aller im Cache gespeicherten Projekte. Die Liste zeigt den Namen des Benutzers, der das Projekt zuletzt in Bearbeitung hatte, den Projektnamen, den aktuellen Status und die ID des aktuellen Projekt–Servers. Die Bedeutung der einzelnen Stati entnehmen Sie der folgenden Tabelle:

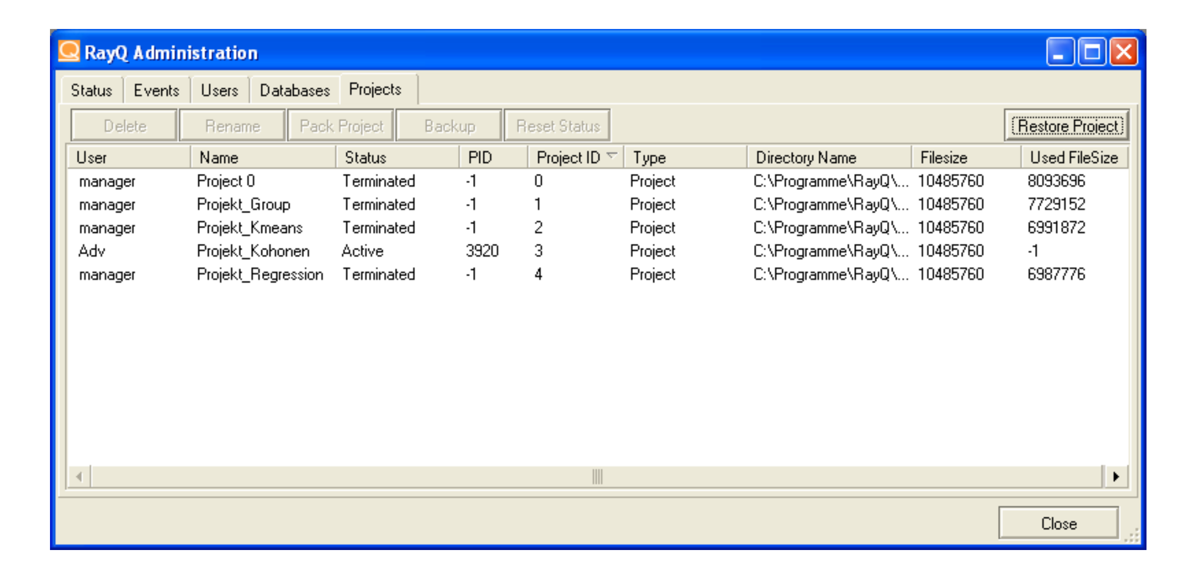


Abbildung 5.21.: Liste aller im Cache gespeicherten Projekte

Status	**Bedeutung**
Terminated	Projekt ist gespeichert
Active	Projekt ist aktuell in Bearbeitung durch einen Client
Archive	Im Archiv gespeichertes Projekt
Nohook	Projekt wird im Hintergrund ausgeführt, es ist zur Zeit kein Client aktiv

Tabelle 5.1.: Projekt–Stati

KAPITEL 6

RayQ Server

Der RayQ Server verwaltet den Daten–Cache und somit alle von den Benutzern angelegten Analyseprojekte. Mit jedem geöffneten Projekt wird ein eigener Server–Prozess gestartet, der alle Berechnungen ausführt.

6.1. Programmsteuerung

6.1.1. Starten

Starten Sie auf dem Rechner, auf dem der Server installiert ist, den „RayQ Administration Server“ *RQAS* aus dem RayQ – Verzeichnis.

> **Wichtig!**
> Ein gestarteter RQAS ist Voraussetzung für das Verwenden von RayQ.

Der gestartete Server ist betriebsbereit, wenn das blaue RayQ–Symbol im Task–Tray erscheint. Sind bereits RayQ–Projekte auf

dem System gespeichert, kann sich dieser Vorgang verzögern, da beim Start alle vorhandenen Projekte indiziert werden.

6.1.2. Beenden

Klicken Sie mit der rechten Maustaste auf das Programmsymbol im System–Tray und wählen Sie im Kontextmenü Exit. Der Server wird daraufhin beendet. Sind noch RayQ–Projekte aktiv — entweder in einem geöffneten RayQ Designer oder als Hintergrundprozess — wird das Beenden des Servers abgebrochen.

6.1.3. Server anhalten

Möchten Sie den Server vorübergehend anhalten, klicken Sie mit der rechten Maustaste auf das Programmsymbol im System–Tray und wählen im Kontextmenü *Stop Server*. Während der Server angehalten ist, können keine Benutzer angemeldet oder Projekte geöffnet werden.

Zum Fortsetzen der Ausführung wählen Sie *Start Server*.

Das Anhalten ist nicht möglich, so lange RayQ Projekte offen sind oder ausgeführt werden.

6.2. Konfiguration

Klicken Sie mit der rechten Maustaste auf das Server – Programmsymbol in der Task–Leiste, öffnet sich ein Kontextmenü. Wählen Sie *Properties*, um den Konfigurationsdialog aufzurufen.

Vor jedem der Einträge erscheint ein farbiges Symbol. Grün gibt an, dass die jeweilige Einstellung gültig ist, rot dass hier ein Fehler aufgetreten ist.

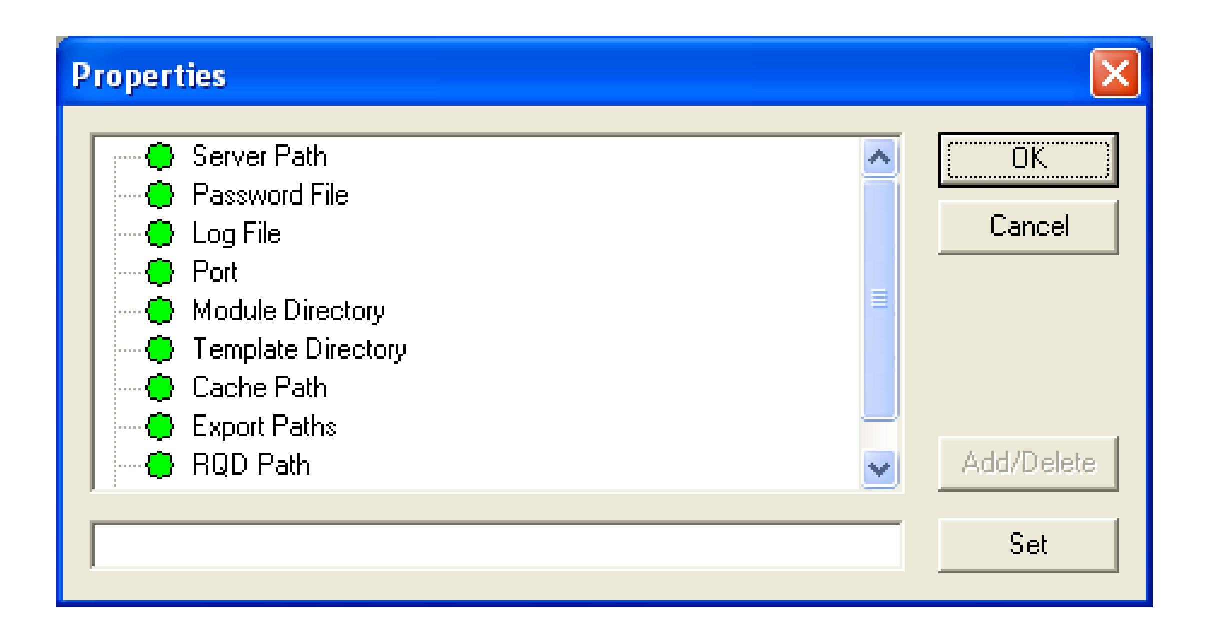


Abbildung 6.1.: Server–Konfiguration

Zum Ändern eines Eintrags klicken Sie auf die betreffende Zeile. Der Wert des Eintrags erscheint darauf hin in dem unter der Tabelle liegenden Eingabefeld und kann von Ihnen bearbeitet werden. Zum Speichern der Änderungen klicken Sie auf *Set*.

Hinweis!
Eine Änderung der Einstellungen ist erst nach einem Neustart des Servers aktiv. Bis dahin arbeitet der Server noch mit den vorherigen Werten.

Wichtig!
Änderungen an der Konfiguration sollten nur von RayQ–Administratoren vorgenommen werden, da falsche Einstellungen den Betrieb beeinträchtigen oder unmöglich machen können.

Parameter	Bedeutung
Cashe	Gibt an, dass der Cache korrekt initalisiert werden konnte
Cache Path	Pfad zum Cache–Verzeichnis
Template Directory	Gibt den Pfad an, in dem Container (siehe 4.7 Templates auf Seite 191) abgespeichert werden
Module Directory	Gibt den Pfad an, in dem die Modul–Bibliotheken (siehe 2.1.4 Plug–In–Module auf Seite 21) gespeichert werden.
Port	Gibt den TCP/IP–Port an, über den der Server im Netzwerk erreichbar ist (siehe 1.1.1 Kommunikationswege sicherstellen auf Seite 9)
Log File	Gibt den Pfad und Dateinamen des Server–Protokolls an
Password File	Gibt den Pfad und Dateinamen der Benutzerdatenbank an (siehe 5.2.3 Users“ auf Seite 221)
Server Path	Gibt den Pfad und Dateinamen des Server–Dienstprogramms an
Export Paths	Gibt einen oder mehrere Verzeichnispfade an, in die mit dem Modul „Server Export“ (siehe 4.6.3, Seite 189) Daten exportiert werden können. Im RayQ Designer wird nur die Bezeichnung des Pfades angezeigt, nicht aber den tatsächlichen Speicherort. Um einen neuen Pfad einzufügen, klicken Sie *Add*. Um die Bezeichnung des Pfades zu ändern, wählen Sie ihn mit einem verzögerten zweiten Mausklick aus.

Tabelle 6.1.: Eigenschaften des Servers

Anhang

ANHANG A

Technische Daten

A.1. Systemanforderungen

A.1.1. RayQ Server

Minimalausstattung

Windows 2000 Professional

400 MHz Prozessor

256 MB RAM

50 MB freier Festplattenspeicher für Programmumgebung

1 GB freier Festplattenspeicher für Daten- und Projekt–Cache

Empfohlene Ausstattung

Windows 2000 Server

1 GHz Multiprozessor–System

1 GB RAM

50 MB freier Festplattenspeicher für Programmumgebung

10 GB freier Festplattenspeicher für Daten- und Projekt–Cache

A.1.2. RayQ Client

Minimalausstattung

Windows 2000 Professional

300 MHz Prozessor

128 MB RAM

10 MB freier Festplattenspeicher für Programmumgebung

Grafikkarte mit 16 Bit Farbtiefe und Auflösung von 1024x768

Empfohlene Ausstattung

Windows 2000 Professional

1 GHz Prozessor

256 MB RAM

10 MB freier Festplattenspeicher für Programmumgebung

Grafikkarte mit 3D–Beschleunigung (OpenGL), 32 Bit Farbtiefe und Auflösung von 1280x1024

ANHANG B

Unterstützung

B.1. Fehlerbehebung

B.1.1. Installation

Das Setup–Programm konnte einzelne Dateien nicht kopieren, das Update wurde abgebrochen.

Prüfen Sie, ob ein RayQ–Programm zur Zeit auf dem betreffenden Rechner aktiv ist. Beenden Sie alle Programme oder starten den Rechner gegebenenfalls neu.

B.1.2. RayQ Designer

Table View

Es werden keine Daten in der Tabelle angezeigt.

1. Die Analyse wurde nicht gestartet — das ist übrigens auch dann notwendig, wenn Sie sich mit einer Table–View nur den Inhalt einer Datenquelle anzeigen lassen und zunächst keine weitere Berechnung vornehmen möchten.

2. Es wurde die falsche Tabelle ausgewählt. Stellen Sie sicher, dass in der Auswahlbox am oberen Rand die gewünschte Tabelle ausgewählt wurde. Gegebenenfalls gibt es auch mehrere Tabellen gleichen Namens.

3. Es ist kein oder ein fehlerhaftes SQL–Statement angegeben, wodurch keine Daten aus der Datenbank selektiert wurden.

4. Es wurden widersprüchliche Bedingungen in einem Datasource- oder Join–Modul angegeben.

Graphic View

Beim Öffnen einer Graphic View erscheint nur ein Ausschnitt der Grafik innerhalb des Fensters.

Installieren Sie den aktuellen Grafiktreiber des Herstellers der in den Client eingebauten Grafikkarte. Die bei Windows 2000 und XP standardmäßig mitgelieferten Treiber unterstützen nur einen Teil der OpenGL–Befehle, so dass es zu Problemen bei der korrekten Darstellung kommen kann.

Beim Öffnen einer Graphic View erscheint eine Fehlermeldung, dass keine OpenGL–Grafik unterstützt wird.

Der Graphic–View nutzt die OpenGL–Schnittstelle, um dreidimensionale Objekte darstellen zu können. Aus diesem Grund ist es sehr empfehlenswert, eine Grafikkarte zu verwenden, die eine hardwareseitige 3D–Beschleunigung zur Verfügung stellt. Um den vollen Leistungsumfang der Grafikkarte nutzen zu können, ist in der Regel die Installation der aktuellen Treibersoftware des Herstellers notwendig.

Steht keine Hardware–Unterstützung der 3D–Funktionen zur Verfügung, wird die Darstellung von der Software simuliert. Diese Variante ist deutlich langsamer und daher nicht empfehlenswert.

Lange Ladezeiten beim Öffnen von Fenstern

Das Öffnen eines View–Fensters dauert ungewöhnlich lang, obwohl der PC schnell genug sein sollte und der neueste Grafiktreiber installiert ist.

Dieses Verhalten tritt im Zusammenhang mit dem Programm „NView“ von „NVidia“ auf, das zusätzliche Funktionen in Windows – Programmfenster integrieren soll. Deaktivieren Sie *NView* und entfernen es aus dem Systemstart (Registry).

Eine im Vergleich zu anderen Fenstern längere Wartezeit beim Öffnen des Graphic View ist hingegen normal, da hier eine OpenGL–Ansicht mit einer Vielzahl von Objekten initialisiert werden muss.

ANHANG C

Funktionen

C.1. Allgemeine Hinweise

C.1.1. Genauigkeit

Alle Berechnungen werden auf Basis der Fließkomma–Arithmetik des Prozessors durchgeführt. Bitte beachten Sie, dass die Genauigkeit der Berechnungen aus diesem Grund begrenzt ist. Mehrere aufeinander folgende Berechnungen mit großen Wertebereichen können den systembedingten Fehler des Prozessors potenzieren.

C.1.2. Positionen

Zeichenpositionen innerhalb von Zeichenfolgen (Strings) werden ab dem Index „0“ beginnend gezählt. Das erste Zeichen hat demnach die Position „0“, das zweite „1“. Die Position des letzten Zeichens entspricht der Länge der Zeichenfolge minus 1.

C.1.3. Bogenmaß

Die trigonometrischen Funktionen werden auf Basis des Bogenmaßes bestimmt. Um eine Gradzahl in das Bogenmaß und um-

gekehrt zu wandeln, verwenden Sie bitte die Funktionen Radian bzw. Degree (siehe Seite 254).

C.1.4. Datum

Alle Datumswerte beziehen sich auf den 15.10.1582 um 0 Uhr, den Beginn des gregorianischen Kalenders. Dieses Datum entspricht dem Zahlenwert „0“, ab dem sekundenweise gezählt wird. Der erste Tag nach diesem Null–Datum entspricht demnach einem Wert von „86400“.

C.2. Operatoren

RayQ hat drei Arten von Operatoren: Arithmetische Operatoren, Vergleichsoperatoren und logische Operatoren.

C.2.1. Arithmetische Operatoren

Als arithmetische Operatoren stehen die vier Grundrechenarten zur Verfügung.

Addition

Addiert zwei Operanden miteinandander.

Operator	Beispiel
+	3 + 5 Ergebnis: 8

Subtraktion

Subtrahiert den zweiten Operanden von dem ersten.

Operator	Beispiel
-	7 - 4 Ergebnis: 3

Multiplikation

Multipliziert zwei Operanden miteinander.

Operator	Beispiel
*	5 * 4 Ergebnis: 20

Division

Teilt den ersten Operanden durch den zweiten.

Operator	Beispiel
/	15 / 3 Ergebnis: 5

C.2.2. Vergleichsoperatoren (Comparison)

Vergleichsoperatoren liefern einen Wahrheitswert zurück. Ist die Aussage des Vergleiches falsch, wird der Wert 0 zurückgeliefert, ist sie wahr, 1.

Gleich

Vergleicht zwei Operanden auf einen identischen Wert.

Operator	Beispiel
=	3 = 3 Ergebnis: 1

Größer gleich

Stellt fest, ob der erste Operand im Vergleich zum zweiten einen größeren oder identischen Wert besitzt.

Operator	Beispiel
>=	3 >= 3 Ergebnis: 1

Größer

Stellt fest, ob der erste Operand im Vergleich zum zweiten einen größeren Wert besitzt.

Operator	Beispiel
>	3 > 3 Ergebnis: 0

Kleiner gleich

Stellt fest, ob der erste Operand im Vergleich zum zweiten einen kleineren oder identischen Wert besitzt.

Operator	Beispiel
<=	3 <= 3 Ergebnis: 1

Kleiner

Stellt fest, ob der erste Operand im Vergleich zum zweiten einen kleineren oder identischen Wert besitzt.

Operator	Beispiel
<	3 < 3 Ergebnis: 0

Ungleich

Stellt fest, ob zwei Operanden einen unterschiedlichen Wert besitzen.

Operator	Beispiel
<>	3 <> 3 Ergebnis: 0
!=	4 != 3 Ergebnis: 1

Ist

Dieser Operator kann ausschließlich für den Vergleich mit dem von der konstanten Funktion *Null()* zurückgelieferten NULL–Wert verwendet werden.

Operator	Beispiel
IS	x IS NULL() Ergebnis: 1, falls x dem Wert „Null“ entspricht. Ergebnis: 0, falls x nicht dem Wert „Null“ entspricht.

Textvergleich mit Wildcards

Der Like–Operator erlaubt einen Mustervergleich innerhalb von Texten. Für den Aufbau der Muster stehen folgende Joker–Zeichen („Wildcards“) zur Verfügung:

% steht für eine beliebige Anzahl beliebiger Zeichen,
_ steht für ein einzelnes beliebiges Zeichen.

Möchten Sie nach einem Zeichen suchen, das einem dieser Wildcards entspicht, schreiben Sie \% oder _, um deren Sonderfunktion auszuklammern.

Syntax	Parameter
s1 LIKE *s2*	*s1* ist der Text, der mit dem Muster *s2* verglichen werden soll.
Beispiel	**Ergebnis**
Hallo Welt! LIKE _allo%	1
1.0% LIKE _ . _\%	1
1.0% Anteil LIKE _._\%	0
1.0% Anteil LIKE _._\%%	1

C.2.3. Logische Operatoren (Logical)

Logische Operatoren liefern einen Wahrheitswert zurück. „Falsch“ wird durch den Wert 0 symbolisiert, „wahr“ durch den Wert 1. Umgekehrt wird der Zahlenwert 0 als „falsch“ interpretiert, jeder andere als „wahr“.

Für logische Verknüpfungen stehen die folgenden Operatoren zur Verfügung:

UND

Operator	Beispiel
AND	1 AND 0 Ergebnis: 0 1 AND 1 Ergebnis: 1

ODER

Operator	Beispiel
OR	1 OR 0 Ergebnis: 1 1 OR 1 Ergebnis: 1

Exklusives ODER

Operator	Beispiel
XOR	1 XOR 0 Ergebnis: 1 1 XOR 1 Ergebnis: 0

Negation

Operator	Beispiel
NOT	NOT 1 Ergebnis: 0
!	! 0 Ergebnis: 1

Hinweis!
Die Schreibweise $NOT()$ ist zulässig, stellt jedoch keine Funktion dar, sondern nur die logische Gruppierung der in den Klammern stehenden Operanden.

C.3. Funktionen

Funktionen werden durch eine Funktionsbezeichnung und einer Parameterliste in geschlossenen Klammern bezeichnet. Bitte beachten Sie, dass Funktionen, die keine Parameter erwarten, ebenfalls ein geschlossenes Klammernpaar „()“ enthalten.

C.3.1. Logische Funktionen

Wahrheitsprüfung

Überprüft eine Bedingung und gibt als Ergebnis einen wählbaren Wert für wahre oder falsche Aussagen zurück.

Syntax	Parameter
$IfElse\,(x,a,b)$	x ist die zu prüfende Bedingung. a ist der Wert, der im Falle einer wahren Aussage zurückgegeben wird. a ist der Wert, der im Falle einer falschen Aussage zurückgegeben wird. Haben a und b unterschiedliche Datentypen, wird einheitlich der Typ des Parameters b verwendet.
Beispiel	**Ergebnis**
Sei $x = 7$	
$IfElse\,(x = 1, 10, 20)$	10
$IfElse\,(x > 10, JA, NEIN)$	NEIN

Überprüfung auf NULL

Syntax	Beispiel
$IsNull\,()$	Sei $x = NULL$. $Isnull\,(x)$ Ergebnis: 1 Sei $x = 5$. $Isnull\,(x)$ Ergebnis: 0

> **Hinweis!**
> $IsNull$ ist eine Funktion, die den Inhalt der Klammer auf NULL überprüft und kann nicht als Operator verwendet werden.

C.3.2. Standard–mathematische Funktionen

Bitte beachten Sie die allgemeinen Hinweise ab Seite 245.

Sinus

Berechnet den Sinus eines Wertes α.

Syntax	Beispiel
$Sin\,(\alpha)$	$Sin\left(\frac{Pi}{2}\right)$ Ergebnis: 1

Cosinus

Berechnet den Cosinus eines Wertes α.

Syntax	Beispiel
$Cos\,(\alpha)$	$Cos\left(\frac{Pi}{2}\right)$ Ergebnis: 0

Tangens

Berechnet den Tangens eines Wertes α.

Syntax	Beispiel
$Tan\,(\alpha)$	$Tan\,(Pi)$ Ergebnis: 0

Arcussinus

Berechnet den Arcussinus eines Wertes α.

Syntax	Beispiel
$ArcSin\,(\alpha)$	$ArcSin\,(1)$ Ergebnis: $\left(\frac{\pi}{2}\right)$

Arcuscosinus

Berechnet den Arcuscosinus eines Wertes α.

Syntax	Beispiel
$ArcCos\,(\alpha)$	$ArcCos\,(0)$ Ergebnis: $\left(\frac{\pi}{2}\right)$

Arcustangens

Berechnet den Arcustangens eines Wertes α.

Syntax	Beispiel
$ArcTan\,(\alpha)$	$ArcTan\,(1)$ Ergebnis: $\left(\frac{\pi}{4}\right)$

Sinus Hyperbolicus

Berechnet den Sinus Hyperbolicus eines Wertes α.

Syntax	Beispiel
$Sinh\,(\alpha)$	$Sinh\,(1)$ Ergebnis: ~ 1.175

Cosinus Hyperbolicus

Berechnet den Cosinus Hyperbolicus eines Wertes α.

Syntax	Beispiel
$Cosh\,(\alpha)$	$Cosh\,(1)$ Ergebnis: ~ 1.543

Tangens Hyperbolicus

Berechnet den Tangens Hyperbolicus eines Wertes α.

Syntax	Beispiel
$Tanh\,(\alpha)$	$Tanh\,(1)$ Ergebnis: ~ 0.762

Exponentialfunktion zur Basis e

Syntax	Beispiel
$Exp\,(x)$	$Exp\,(2)$ Ergebnis: ~ 7.389

Natürlicher Logarithmus

Berechnet den natürlichen Logarithmus des Wertes x.

Syntax	Beispiel
$Log\,(x)$	$Log\,(2.7172)$ Ergebnis: 1

Logarithmus zur Basis 10

Berechnet den Logarithmus zur Basis 10 des Wertes x.

Syntax	Beispiel
$Log10\,(x)$	$Log10\,(1000)$ Ergebnis: 3

Potenz

Liefert zur Eingabe von Basis x und Exponent y die dazugehörige Potenz x^y.

Syntax	Beispiel
$Pow\,(x, y)$	$Pow\,(5, 3)$ Ergebnis: 125

Wurzel

Berechnet die Wurzel einer Zahl x.

Syntax	Beispiel
$Sqrt\,(x)$	$Sqrt\,(64)$ Ergebnis: 8

Minimum

Liefert das Minimum der als Parameter gelieferten Werte.

Syntax	Parameter
$Min\,(x_1, x_2, \ldots, x_n)$	Sie können beliebig viele Parameter verwenden, die beliebige Datentypen enthalten können.
Beispiel	**Ergebnis**
$Min\,(4, -3, 1)$	-3
$Min\,(K, S, B)$	B

Maximum

Liefert das Maximum der als Parameter gelieferten Werte.

Syntax	Parameter
$Min\,(x_1, x_2, \ldots, x_n)$	Sie können beliebig viele Parameter verwenden, die beliebige Datentypen enthalten können.
Beispiel	**Ergebnis**
$Min\,(4, -3, 1)$	4
$Min\,(K, S, B)$	S

Absolutwert

Liefert den Betrag einer Zahl x, also den Zahlenwert ohne Vorzeichen.

Syntax	Beispiel
$Abs\,(x)$	$Abs\,(-6)$ Ergebnis: 6

Gradfunktion

Wandelt Bogen- in Gradmaß.

Syntax	Beispiel
$Degree\,(x)$	$Degree\,(pi\,())$ Ergebnis: 180

Bogenmaß

Wandelt Grad- in Bogenmaß.

Syntax	Beispiel
$Radian\,(x)$	$Radian\,(180\,())$ Ergebnis: π

Rundung

Rundet eine Zahl x auf eine bestimmte Anzahl Dezimalstellen.

Syntax	Parameter
$Round\,(x, n)$	x bezeichnet den zu rundenden Wert. n gibt die Anzahl der Dezimalstellen nach dem Komma an, auf die gerundet werden soll. Ist n negativ, so wird auf die entsprechende Stelle vor dem Komma gerundet.
Beispiel	**Ergebnis**
$Round\,(635.927, 2)$	635.93
$Round\,(635.927, -2)$	640

Fakultät

Berechnet die Fakultät einer Ganzzahl x.

Syntax	Parameter
$Factorial\,(x)$	n muss eine natürliche Zahl sein.
Beispiel	**Ergebnis**
$Factorial\,(4)$	$4 \cdot 3 \cdot 2 \cdot 1 = 24$

Modulo

Rest einer ganzzahligen Division.

Syntax	Parameter
$Mod\,(x, y)$	x ist die Zahl, für die der Rest einer Division gesucht wird. y ist die Zahl, durch die x dividiert werden soll.
Beispiel	**Ergebnis**
$Mod\,(15, 4)$	3

Division ohne Rest

Syntax	Parameter
$Div\,(x, y)$	x ist die Zahl, die geteilt werden soll. y ist der Divisor.
Beispiel	**Ergebnis**
$Mod\,(18, 5)$	3

Pi - π

Diese Funktion gibt den Wert Pi zurück.

Syntax	Beispiel
$Pi\,()$	$Pi\,()$ Ergebnis: $\sim$ 3.141

e–Funktion

Berechnet die Eulersche Zahl e.

Syntax	Beispiel
$e\,()$	$e\,()$ Ergebnis: $\sim$ 2.718

Zufallszahl

Erzeugt eine Zufallszahl im angegebenen Wertebereich.

Syntax	Parameter
$Rand\,(a, b)$	a und b bezeichnen die Grenzen des Wertebreichs, in dem die Zufallszahl liegen soll. Dabei ist es irrelevant, ob $a > b$ oder $b > a$.
Beispiel	**Ergebnis**
$Rand\,(1, 4.5)$	2.65
$Rand\,(''a'','' z'')$	$''s''$
$Rand\,(''anfang'','' ende'')$	$''fortschritt''$

C.3.3. String–Funktionen

Bitte beachten Sie die allgemeinen Hinweise ab Seite 245.

Textlänge

Ermittelt die Länge des übergebenen Texts.

Syntax	Beispiel
$StrLen\,(s)$	$StrLen\,("Hallo\,Welt")$ Ergebnis: 10

Hinweis!
Leerzeichen werden auch als Zeichen betrachtet.

Teilstrings

Gibt einen definierten Teil des übergebenen Texts zurück.

Syntax	Parameter
$SubString\,(s, n_1, n_2)$	s ist der Text, aus dem ein Teil extrahiert werden soll. n_1 ist die Position des ersten Zeichens im String s, ab dem ausgeschnitten werden soll. Ist $n_1 < 0$, dann wird n_1 als 0 interpretiert. Ist n_1 größer oder gleich der Länge der Zeichenfolge, gibt die Funktion einen leeren String zurück. n_2 ist die Position des ersten Zeichens nach der auszuschneidenden Passage. Sollte n_2 größer oder gleich der Länge der Zeichenfolge, wird immer bis zum letzten Zeichen ausgeschnitten. Ist n_2 kleiner als n_1, wird ein leerer String zurückgegeben.
Beispiel	**Ergebnis**
$SubString\,("Hallo\,Welt", 6, 10)$	$"Welt"$

Finden

Findet eine Zeichenfolge innerhalb eines Texts und gibt die Position des ersten Zeichens der Zeichenfolge zurück. Ist die Suche erfolglos geblieben, wird der Wert „-1“ zurückgegeben.

Syntax	Parameter
$Find\,(s, s_1\,[, n])$	s ist der Gesamttext. s_1 ist die zu suchende Zeichenfolge. n ist ein optionaler Parameter und gibt die Position an, ab der in s gesucht werden soll. Ist $n < 0$, dann wird am Anfang der Zeichenfolge begonnen. Ist n größer oder gleich der Länge von s, wird „-1“ zurückgegeben
Beispiel	Ergebnis
$Find\,(''Hallo\,Welt'', ''Welt'', 0)$	6

Ersetzen

Ersetzt einen Teil oder den gesamten Text.

Syntax	Parameter
$Replace\,(s_1, s_2, s_3)$	s_1 ist der Gesamttext, in dem ein Teil ersetzt werden soll. s_2 ist die Zeichenfolge, die ersetzt werden soll. s_3 ist die Zeichenfolge, die statt s_2 eingefügt
Beispiel	Ergebnis
$Replace\,(''Hallo\,Welt'', ''Welt'', ''Erde'')$	$''Hallo\,Erde''$

Großbuchstaben

Wandelt alle Zeichen eines Texts (A bis Z, keine Umlaute) in Großbuchstaben um.

Syntax	Parameter
$Upper\,(s)$	s zu wandelnde Zeichenfolge.
Beispiel	Ergebnis
$Upper\,(''Hallo\,Welt'')$	$''HALLO\,WELT''$

Kleinbuchstaben

Wandelt alle Zeichen eines Texts (A bis Z, keine Umlaute) in Kleinbuchstaben um.

Syntax	Parameter
$Lower\,(s)$	s zu wandelnde Zeichenfolge.
Beispiel	Ergebnis
$Lower\,(''Hallo\,Welt'')$	$''hallo\,welt''$

Verketten

Fügt zwei Zeichenfolgen zu einer neuen zusammen.

Syntax	Parameter
$Cat\,(s_1, s_2)$	s_1 und s_2 sind die zwei Zeichenfolgen, die verbunden werden.
Beispiel	Ergebnis
$Cat\,("Hallo","Welt")$	$"HalloWelt"$

Zeichenwert

Liefert den Integer–Zahlenwert des übergebenen Zeichens nach der ASCII–Tabelle zurück.

Syntax	Parameter
$Ascii\,(s)$	Als s kann eine beliebige Zeichenfolge übergeben werden, jedoch wird nur der erste Buchstabe für die Wandlung verwendet.
Beispiel	Ergebnis
$Ascii\,("Test")$	84

Reinigen

Löscht alle Whitespace–Zeichen aus einem Text. Als „Whitespaces“ bezeichnet man die Zeichen für Zeilenumbruch (CR), Zeilenvorschub (LF), Tablulator und das Leerzeichen.

Syntax	Parameter
$Clear\,(s)$	s ist der zu bereinigende Text.
Beispiel	Ergebnis
$Clear\,("\ Hallo\ \ Welt")$	$"HalloWelt"$

Säubern

Löscht alle nicht druckbaren Zeichen aus einem Text. Als „nicht druckbar“ werden alle Zeichen angesehen, die einen ASCII–Wert niedriger als 20 (Leerzeichen) besitzen, also alle so genannten Steuerzeichen.

Syntax	Parameter
$Clean\,(s)$	s ist der zu bereinigende Text.
Beispiel	Ergebnis
$Clean\,("\ Hallo\ \ (Zeilenumbruch)\ \ Welt")$	$"Hallo\,Welt"$

Links abschneiden

Entfernt alle führenenden Whitespace–Zeichen.

Syntax	Parameter
$LTrim\,(s)$	s ist der zu bereinigende Text.
Beispiel	Ergebnis
$LTrim\,(''\quad Hallo\,Welt'')$	$''Hallo\,Welt''$

Rechts abschneiden

Entfernt alle Whitspace–Zeichen am Ende eines Texts.

Syntax	Parameter
$RTrim\,(s)$	s ist der zu bereinigende Text.
Beispiel	Ergebnis
$RTrim\,(''Hallo\,Welt\quad '')$	$''Hallo\,Welt''$

Wiederholung

Füllt einen Text mit einer wiederholten Zeichenfolge.

Syntax	Parameter
$Repeat\,(s, n)$	s ist die Zeichenfolge, die wiederholt werden soll. n ist die Anzahl der Wiederholungen.
Beispiel	Ergebnis
$Repeat\,(''Ha'', 3)$	$''HaHaHa''$

C.3.4. Explizite Typkonvertierungen

Bitte beachten Sie die allgemeinen Hinweise ab Seite 245.

Zeichenfolge

Wandelt einen beliebigen Datentyp in einen String um.

Syntax	Parameter
$AsString\,(x)$	x ist ein Wert beliebigen Typs, der in einen String gewandelt wird.
Beispiel	Ergebnis
$AsString\,(4.56)$	$''4.56''$

Zahlenwert

Wandelt einen beliebigen Datentyp in einen Double–Wert um.

Bei der Konvertierung String nach Double wird ein Punkt als Trennzeichen für die Dezimalstellen erwartet.

Ist der zu wandelnde Wert ein String, kann nur dann eine Wandlung erfolgen, wenn ein Zahlenwert innerhalb der Zeichenfolge erkannt wird. Ansonsten wird 0 zurückgeliefert.

Ein als Zeichenfolge ausgedrücktes Datum kann mit dieser Funktion nicht direkt in eine Zahl umgewandelt werden. Hier ist es notwendig, das Datum mit der Funktion AsDateTime zunächst in den Datumstypen umzuwandeln.

Die Zeichenfolge wird bis zum ersten nicht umwandelbaren Zeichen untersucht und die bis dorthin erkannte Zahl zurückgegeben.

Syntax	Parameter
$AsNumber\,(x)$	x ist ein Wert beliebigen Typs, der in Double gewandelt wird.
Beispiel	**Ergebnis**
$AsNumber\,(''4.56'')$	4.56
$AsNumber\,(''Hallo'')$	0.0
$AsNumber\,(''Hallo\ 32'')$	32.0
$AsNumber\,(''25.05.2010'')$	25.05

Datum

Wandelt einen Text oder eine Zahl in einen Datumswert, sofern er einem der folgenden Formate entspricht:

- TT.MM.JJ hh:mm:ss
- TT.MM.JJJJ hh:mm:ss
- TT-MMM-JJ hh:mm:ss
- TT-MMM-JJ hh:mm:ss
- TT-MMM-JJJJ hh:mm:ss
- MM/TT/JJ hh:mm:ss
- JJJJ.MM.TT hh:mm:ss
- JJJJ-MM-TT hh:mm:ss
- TT.MM.JJ

- TT.MM.JJJJ
- TT-MMM-JJ
- TT-MMM-JJJJ
- MM/TT/JJ
- MM/TT/JJJJ
- JJJJ.MM.TT
- JJJJ-MM-TT
- JJJJMMTT
- JJJJMMTThhmmss

„TT“ bezeichnet den zweistelligen Tag, „MM“ den zweistelligen Monat, „MMM“ den Monat in Textform (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC), „JJ“ die letzten zwei Ziffern des Jahres, „JJJJ“ die volle, vierstellige Jahreszahl, „hh“, „mm“, „ss“ stehen jeweils für die zweistelligen Stunde, Minuten und Sekunden.

Syntax	Parameter
$AsDateTime\,(x)$	x ist ein Wert beliebigen Typs, der in Datetime gewandelt wird. Enthält x nur ein Datum, so wird als Uhrzeit immer 0 Uhr angenommen.
Beispiel	**Ergebnis**
$AsDateTime\,(''12.08.2005'')$	$12.08.2005\ \ 00:00:00$
$AsDateTime\,(24)$	$15.10.1582\ \ 00:00:24$

Aktuelle Zeile

Liefert die Nummer des aktuellen Datensatzes der Tabelle, auf dem die Berechnung stattfindet. Hiermit kann zum Beispiel ein künstlicher Schlüsselwert erzeugt werden. Bitte beachten Sie, dass die Numerierung der Zeilen bei 0 beginnt.

Syntax	Beispiel
$CurrentRow\,()$	$CurrentRow\,()$ Ergebnis: z.B. 15

Erste Zeile

Liefert die Nummer der ersten Zeile der Tabelle.

Syntax	Beispiel
$FirstRow\,()$	$FirstRow\,()$ Ergebnis: 0, da die erste Zeile einer Tabelle immer mit „0“ bezeichnet wird)

Letzte Zeile

Liefert die Nummer der ersten Zeile der Tabelle.

Syntax	Beispiel
$LastRow\,()$	$LastRow\,()$ Ergebnis: 1234, wenn die Tabelle 1233 Zeilen enthält.

NULL

Liefert einen NULL–Wert zurück, um z.B. in das Ergebnis einer Funktion auf NULL zu überprüfen.

Syntax	Beispiel
$Null\,()$	$''Spaltenname''\ IS\ Null\,()$ Ergebnis: 1, wenn in der aktuelle Zeile ein NULL–Wert steht

Hinweis! Die konstante Funktion $Null\,()$ darf nur in Vergleichen verwendet werden, nicht innerhalb von Funktionstermen. Deshalb kann $Null\,()$ nur zusammen mit dem Operator „IS“ verwendet werden, nicht aber mit „=“.

C.3.5. Gruppierungsfunktionen (Special Group Functions)

Heaviside

Berechnet die Heavisidefunktion, die für alle Werte kleiner 0 eine 0 zurückliefert und für alle Werte größer oder gleich 0 eine 1.

Syntax	Parameter
$Heaviside\,(x)$	x ist der zu prüfende Wert.
Beispiel	Ergebnis
$Heaviside\,(4)$	1
$Heaviside\,(-3.56)$	0

Chi–Funktion

Berechnet die Chi–Funktion, die für alle Werte innerhalb eines gegebenen Intervalls 1 zurückliefert und ansonsten 0.

Syntax	Parameter
$Chi\,(x, a, b)$	x ist der zu testende Wert. a ist die untere Grenze des Intervalls. b ist die obere Grenze des
Beispiel	Ergebnis
$Chi\,(3, 5, 10)$	0
$Chi\,(7, 5, 10)$	1

Treppenfunktion

Erlaubt Werte anhand verschiedener Intervalle einen anderen Wert zuzuweisen.

Syntax	Parameter
$Stair\,(x, a_1, b_1, \ldots, v_1, \ldots, a_n, b_n, \ldots, v_n)$	Die Parameter-Anzahl ist immer $1+n\cdot 3$. x ist der zu testende Wert. Die darauf folgenden Dreiergruppen a_i, b_i, v_i legen das Intervall und den Wert fest, der zurückgegeben werden soll, wenn sich x im jeweiligen Intervall befindet. a_1 bis a_n sind die kleinsten Werte des jeweiligen Intervalls. b_1 bis b_n sind die jeweils größten Werte eines Intervalls. v_1 bis v_n sind die Werte, die zurückgeliefert werden, wenn sich x in dem jeweiligen Intervall befindet.
Beispiel	Ergebnis
$Stair\,(2, 0, 1, 1, 1, 3, 4)$	4
$Stair\,(0.5, 0, 1, 5, 1, 6, 7)$	5

Intervallfunktion

Teilt ein Intervall in gleiche Teile und ordnet einem gegeben Wert jeweils den unteren Wert des Intervalls zu, in dem es zu liegen kommt.

Syntax	Parameter
$Digit\,(x, a, b, N)$	x ist der Wert, der einem Intervall zugeordnet werden soll. a ist der kleinste Wert des Intervalls. b ist der größte Wert des Intervalls. N ist die Anzahl der Unterintervalle, in die das Intervall $[a, b]$ eingeteilt werden soll.
Beispiel	**Ergebnis**
$Digit\,(2, 0, 10, 4)$	0
$Digit\,(5.6, 0, 10, 4)$	5

C.3.6. Datums- und Zeitfunktionen (Date & Time Functions)

Bitte beachten Sie die allgemeinen Hinweise ab Seite 245.

Sekunde

Extrahiert aus einem Datetime–Wert die Sekunden.

Syntax	Parameter
$Second\,(d)$	d ist das Datum, dessen Sekunden bestimmt werden sollen.
Beispiel	**Ergebnis**
$Second\,(''24.10.2000\ 12:45:32'')$	32

Minute

Extrahiert aus einem Datetime–Wert die Minute.

Syntax	Parameter
$Minute\,(d)$	d ist das Datum, dessen Minuten bestimmt werden sollen.
Beispiel	**Ergebnis**
$Minute\,(''24.10.2000\ 12:45:32'')$	45

Stunde

Extrahiert aus einem Datetime–Wert die Minute.

Syntax	Parameter
$Hour\,(d)$	d ist das Datum, dessen Stunde bestimmt werden soll.
Beispiel	**Ergebnis**
$Hour\,(''24.10.2000\ 12:45:32'')$	12

Tag

Liefert die Anzahl der Tage ab dem Anfang des gregorianischen Kalenders (15.10.1582). Unter Verwendung einer Subtraktion lassen sich so die Anzahl der Tage zwischen zwei Datumsangaben berechnen.

Syntax	Parameter
$Day\,(d)$	d ist das Datum, dessen Anzahl von Tagen seit dem 0-Datum berechnet werden soll.
Beispiel	**Ergebnis**
$Day\,(''12.03.1967'')$	140401
$Day\,(''12.03.1967'') - Day\,(''10.03.1967'')$	2

Tag eines Monats

Extrahiert aus einem gegebenen Datum den Tag des jeweiligen Monats.

Syntax	Parameter
$DayOfMonth\,(d)$	d ist das Datum, aus dem der Tag des jeweiligen Monats extrahiert werden soll.
Beispiel	**Ergebnis**
$DayOfMonth\,(''24.10.2000\ 12:45:32'')$	24

Tag eines Jahres

Extrahiert aus einem Datum die Anzahl der Tage seit Beginn des jeweiligen Jahres.

Syntax	Beispiel
$DayOfYear\,(d)$	$DayOfYear\,(''12.03.1967\ 12:45:32'')$ Ergebnis: 71

Monat

Extrahiert aus einem Datetime–Wert den Monat.

Syntax	Parameter
$Month\,(d)$	d ist das Datum, aus dem der Monat extrahiert werden soll.
Beispiel	**Ergebnis**
$Month\,(''24.10.2000\ 12:45:32'')$	10

Jahr

Extrahiert aus einem Datetime–Wert das Jahr.

Syntax	Parameter
$Year\,(d)$	d ist das Datum, aus dem das Jahr extrahiert werden soll.
Beispiel	**Ergebnis**
$Year\,(''24.10.2000\ 12:45:32'')$	2000

Heute

Gibt das aktuelle Datum als Datetime–Wert zurück.

Syntax	Beispiel
$Today\,()$	$Today\,()$ Ergebnis: $''09.10.2010\ 00:00:00''$

Jetzt

Gibt die aktuelle Uhrzeit und Datum als Datetime–Wert zurück.

Syntax	Beispiel
$Now\,()$	$Now\,()$ Ergebnis: $''09.10.2010\ 03:32:11''$

Wochentag

Berechnet den Wochentag eines gegebenen Datums, wobei für jeden Wochentag von Montag bis Sonntag die Werte 1 bis 7 zurückgegeben werden.

Syntax	Parameter
$Weekday\,(d)$	d ist der Datetime–Wert, dessen Wochentag bestimmt werden soll.
Beispiel	**Ergebnis**
$Weekday\,(''22.10.2010\ 12:45:32'')$	5

C.3.7. Spaltenfunktionen

Spaltenfunktionen ziehen für ihre Berechnung die Werte aller Zeilen einer Spalte heran. Auf diese Weise lässt sich z.B. in einer Formel der aktuellen Wert einer Zeile vom Durchschnittswert der gesamten Spalte abziehen, um die Abweichungen der Einzelwerte ausgeben zu können.

Wichtig!
Zur Unterscheidung gegenüber zeilenbezogenen Funktionen wird der Spaltenname bei Spaltenfunktionen immer in eckigen Klammern angegeben.

Für die Berechnung der Beispiele auf den folgenden Seiten wird die unten stehende Tabelle verwendet:

Nachname	Alter
Mustermann	45
Becker	13
Schmitt	24
Meier	72
Müller	-

Tabelle C.1.: Beispiel–Tabelle

Anzahl gültiger Werte

Gibt die Anzahl der Zeilen einer Spalte aus, die nicht mit NULL gefüllt sind.

Syntax	Parameter
$ValidRows\left([Spalte]\right)$	$Spalte$ ist die zu verwendende Spalte.
Beispiel	Ergebnis
$ValidRows\left([Alter]\right)$	4

Summe

Berechnet die Summe aller Werte eine Spalte.

Syntax	Parameter
$Sum\left([Spalte]\right)$	$Spalte$ ist die zu verwendende Spalte.
Beispiel	Ergebnis
$Sum\left([Alter]\right)$	154

Durchschnitt

Berechnet die Summe aller Werte eine Spalte.

Syntax	Parameter
$Avg([Spalte])$	$Spalte$ ist die zu verwendende Spalte.
Beispiel	Ergebnis
$Avg([Alter])$	38.5 $= (45 + 13 + 24 + 72) \div 4$

Quadratischer Durchschnitt

Berechnet den Durchschnitt der Quadrate der Einzelwerte einer Spalte.

Syntax	Parameter
$QuadAvg([Spalte])$	$Spalte$ ist die zu verwendende Spalte.
Beispiel	Ergebnis
$QuadAvg([Alter])$	1988.5 $= \left(45^2 + 13^2 + 24^2 + 72^2\right) \div 4$

Standardabweichung

Berechnet die Standardabweichung aller Werte einer Spalte.

Syntax	Parameter
$Deviation([Spalte])$	$Spalte$ ist die zu verwendende Spalte.
Beispiel	Ergebnis
$Deviation([Alter])$	22.5

Varianz

Berechnet die Standardabweichung aller Werte einer Spalte.

Syntax	Parameter
$Variance([Spalte])$	$Spalte$ ist die zu verwendende Spalte.
Beispiel	Ergebnis
$Variance([Alter])$	506.25

Minimum

Berechnet die Standardabweichung aller Werte einer Spalte.

Syntax	Parameter
$Minimum([Spalte])$	$Spalte$ ist die zu verwendende Spalte.
Beispiel	Ergebnis
$Minimum([Alter])$	13

Maximum

Berechnet die Standardabweichung aller Werte einer Spalte.

Syntax	Parameter
$Maximum\,([Spalte])$	$Spalte$ ist die zu verwendende Spalte.
Beispiel	Ergebnis
$Maximum\,([Alter])$	72

ANHANG D

Lizenzvereinbarung

Vertrag

zwischen

Firma Qyte GmbH, Steinbacher Str. 8, D–65760 Eschborn

– nachstehend Lizenzgeber genannt –

und

Lizenznehmer.

D.1. Lizenzgegenstand

Der Lizenzgeber gewährt dem Lizenznehmer eine nicht – ausschließliche und nicht übertragbare Lizenz zur Nutzung des dem Lizenznehmer gelieferten und mit einer Seriennummer versehenen Softwareprogramms betreffend RayQ.

D.2. Umfang der Lizenz

1. Die Lizenz berechtigt den Lizenznehmer, die Software in maschinen – lesbarer Form und die zu seinem Gebrauch notwendigen Unterlagen zu nutzen.

2. Der Lizenznehmer darf die Software auf jeder ihm zur Verfügung stehenden Hardware einsetzen. Wechselt der Lizenznehmer die Hardware, muss er die Software von der bisher verwendeten Hardware löschen. Die Software darf nur auf der Anzahl von Arbeitsplätzen eingerichtet werden, die im Umfang der Lizenz bestimmt ist. Der Lizenznehmer ist berechtigt, zum Zweck der Datensicherung eine Archivkopie anzufertigen. Weitere Vervielfältigungen der Software sowie die Weitergabe des überlassenen Programms an Dritte, gleich aus welchem Grund und gleich welcher Art, sind untersagt. Für jeden Fall der Zuwiderhandlung gegen diese Verpflichtung zahlt der Lizenznehmer dem Lizenzgeber unter Ausschluss des Fortsetzungszusammenhangs eine Vertragsstrafe von EUR 5.000,00.

D.3. Lizenzgebühr

Der Lizenznehmer zahlt an den Lizenzgeber für die Überlassung der Software eine einmalige Lizenzgebühr. Nutzt der Lizenznehmer die Software auf einem Mehrfachplatzsystem, bei dem mehrere Anwender Zugriff auf die Software haben, ist über die zu zahlende Lizenzgebühr eine gesonderte Regelung zu treffen. Der Lizenznehmer ist verpflichtet, den Lizenzgeber von der Mehrfachnutzung in Kenntnis zu setzen.

D.4. Schutz des Lizenzmaterials

1. Der Lizenznehmer ist verpflichtet, den unbefugten Zugriff Dritter auf das Programm sowie die Dokumentation durch geeignete Vorkehrungen zu verhindern.

2. Der Lizenznehmer wird die gelieferten Original–Datenträger

an einem gegen den unberechtigten Zugriff Dritter gesicherten Ort aufbewahren sowie seine Mitarbeiter nachdrücklich auf die Einhaltung der vorliegenden Vertragsbedingungen und des Urheberrechts hinweisen. Insbesondere wird der Lizenznehmer seine Mitarbeiter auffordern, keine unberechtigten Vervielfältigungen des Programms oder des Benutzerhandbuchs anzufordern.

3. Der Lizenznehmer verpflichtet sich, die im Lizenzmaterial enthaltenen Schutzvermerke, wie Copyright–Vermerke und andere Rechtsvorbehalte unverändert beizubehalten sowie alle vom Lizenznehmer hergestellten vollständigen oder teilweisen Kopien von maschinenlesbaren Lizenzmaterial in unveränderter Form zu übernehmen.

D.5. Zusatzleistungen

Darüber hinaus kann der Lizenznehmer folgende, nach Aufwand gesondert zu berechnende Leistungen in Anspruch nehmen:

1. Implementierung der nach Ziff. I übersandten jüngsten Fassung der Standardversion des Software–Produktes.
2. Implementierung der nach Ziff. VII. 2 übersandten Änderungen.
3. Übersendung und Installation von Anpassungen der Software an wesentliche Änderungen gesetzlicher oder sonst verbindlicher Bestimmungen.
4. Beseitigung der nicht vom Lizenzgeber zu vertretenden Fehler.
5. Wiederherstellung der Software auf dem Programmträger bei Beschädigung durch Bedienungs- und/oder Hardware–Fehler.
6. Durchführung von Nachschulungen, die aufgrund der vom Lizenzgeber vorgenommenen oder veranlassten Änderungen oder Verbesserungen notwendig werden.

7. Umstellung der Software auf ein anderes Betriebssystem, ein anderes Hardware–System oder eine andere Programmiersprache, sofern hierfür vom Auftragnehmer eine entsprechende Version angeboten wird.

8. Notwendige Anpassungsarbeiten an der Software bei Änderungen bestehender Betriebssysteme durch deren Hersteller.

9. Beratung in allen Fragen des Einsatzes oder der Anwendung des Software–Produkts einschließlich der Weitergabe von Einsatz– und Anwendungserfahrungen aus dem gesamten Benutzerkreis.

10. Telefonischer Anwendersupport nach Servicelevels.

D.6. Gewährleistung

1. Der Lizenzgeber übernimmt für die Laufzeit dieses Vertrags die Gewähr, dass das Software–Programm vom Lizenznehmer für den bestimmungsgemäßen Zweck benutzt werden kann.

2. Der Lizenzgeber verpflichtet sich, nicht ordnungsgemäß ausgeführte Pflege- und Wartungsleistungen nachzubessern. Nicht unter die Gewährleistungsverpflichtung fallen Fehler, die vom Lizenzgeber nicht zu vertreten sind, wie Fehler infolge von Hardware-Fehlern, Änderungen an der Software des Betriebssystems und der Dateiverwaltung sowie infolge von Fehlbedienung.

3. Soweit nichts anderes bestimmt ist, haftet der Lizenzgeber nicht für Schäden, die nicht an der gelieferten Software selbst entstanden sind; insbesondere übernimmt der Lizenzgeber keine Haftung für Datenverlust oder sonstige Folgeschäden.

4. Kommt der Lizenzgeber der Pflicht zur Mängelbeseitigung innerhalb einer angemessenen Frist nicht nach, kann der Lizenznehmer nach fruchtlosem Ablauf einer angemessenen

Nachfrist und endgültigem Scheitern der Nachbesserung entweder angemessene Herabsetzung der Vergütung oder Rückgängigmachung des Vertrages verlangen. Weitergehende Ansprüche sind ausgeschlossen, gleich aus welchem Rechtsgrund. Dies gilt auch für Ansprüche aus der Verletzung vertraglicher Nebenpflichten. Diese Haftungsbeschränkung gilt nicht, soweit die Ansprüche auf vorsätzlichem oder grob fahrlässigem Handeln des Lizenzgebers oder dem Fehlen von zugesicherten Eigenschaften bzw. Garantiezusagen beruhen.

5. Der Lizenznehmer ist verpflichtet, dem Lizenzgeber nachprüfbare Unterlagen über Art und Auftreten von Abweichungen von der Leistungsbeschreibung zur Verfügung zu stellen und bei der Eingrenzung von Fehlern mitzuwirken.

6. Zugesicherte Eigenschaften sind nur solche, die anhand der Dokumentation zur Software bei Einhaltung der in der Dokumentation beschriebenen Programmbedienung zu den dort beschriebenen Ergebnissen führen.

D.7. Sonstiges

1. Der Lizenznehmer darf seine Rechte aus diesem Vertrag nicht auf Dritte übertragen.

2. Änderungen, Ergänzungen, Kündigungen oder Aufhebungen dieses Vertrags bedürfen der Schriftform.

3. Es gilt deutsches Recht. Soweit zulässig, wird als Gerichtsstand der Sitz des Lizenzgebers vereinbart.

4. Ergänzend zu diesem Vertrag geltend die allgemeinen Geschäftsbedingungen des Lizenzgebers. Der Lizenznehmer erklärt, dass ihm diese bekannt sind.